

# Bibliothek - Entity Framework

## Beziehungen

### Teil 3: Beziehungen in der Bibliotheksverwaltung

In dieser Phase der Implementierung werden die Beziehungen zwischen den verschiedenen Entitäten der Bibliotheksverwaltung definiert. Dabei sollen die wesentlichen Verbindungen zwischen den Klassen abgebildet werden. Die Beziehungen umfassen sowohl Eins-zu-eins (1..1), Eins-zu-viele (1..n) als auch Viele-zu-viele (m..n) Verbindungen.

#### Book und BookDetails (1:1)

In diesem Bibliothekssystem benötigen wir eine genaue Übersicht über die Verfügbarkeit der Bücher. Jede Book-Entität hat eine detaillierte Information über die Anzahl und Verfügbarkeit der Exemplare, die in der Bibliothek existieren.

#### Ziel der Aufgabe

Erstellen Sie eine 1:1-Beziehung zwischen den Klassen Book und BookDetails in Entity Framework Code First. Die BookDetails-Klasse soll die Bestandsinformationen eines bestimmten Buchs festhalten und wird über die BookId-Eigenschaft eindeutig mit einem Buch verknüpft.

#### Anforderungen

1. **BookDetails-Klasse:** Erstellen Sie eine neue Klasse BookDetails mit den folgenden Eigenschaften:
  - **Id:** Primärschlüssel (int) & Fremdschlüssel
  - **TotalCopies:** Gesamtanzahl der Exemplare des Buches in der Bibliothek (int)
  - **BorrowedCopies:** Anzahl der aktuell ausgeliehenen Exemplare (int)
  - **AvailableCopies:** Anzahl der verfügbaren Exemplare (int)
2. **1:1-Beziehung konfigurieren:** Stellen Sie sicher, dass die Beziehung zwischen Book und BookDetails in der Datenbank als 1:1 abgebildet wird.

```
-- BookDetails für das Buch "Das Kapital"
INSERT INTO BookDetails (BookId, TotalCopies, BorrowedCopies, AvailableCopies)
VALUES (1, 10, 2, 8);

-- BookDetails für das Buch "1984"
INSERT INTO BookDetails (BookId, TotalCopies, BorrowedCopies, AvailableCopies)
VALUES (2, 15, 5, 10);

-- BookDetails für das Buch "Einführung in die Programmierung"
INSERT INTO BookDetails (BookId, TotalCopies, BorrowedCopies, AvailableCopies)
VALUES (3, 20, 7, 13);

-- BookDetails für die Biografie "Die Biografie von Albert Einstein"
INSERT INTO BookDetails (BookId, TotalCopies, BorrowedCopies, AvailableCopies)
VALUES (4, 12, 4, 8);

-- BookDetails für das Buch "Der Marsianer"
INSERT INTO BookDetails (BookId, TotalCopies, BorrowedCopies, AvailableCopies)
VALUES (5, 10, 3, 7);

-- BookDetails für das Buch "Harry Potter und der Stein der Weisen"
INSERT INTO BookDetails (BookId, TotalCopies, BorrowedCopies, AvailableCopies)
VALUES (6, 25, 10, 15);

-- BookDetails für das Buch "Der Hund von Baskerville"
INSERT INTO BookDetails (BookId, TotalCopies, BorrowedCopies, AvailableCopies)
VALUES (7, 8, 2, 6);
```

## Beziehung zwischen Author und Book (1..n)

Im Bibliothekssystem werden die Beziehungen zwischen Autoren und Büchern verwaltet. Jeder Autor kann mehrere Bücher schreiben, aber jedes Buch wird nur von einem einzigen Autor verfasst.

Implementiere die Beziehung zwischen Author und Book so, dass:

1. Jedes Buch genau einen Autor hat.
  2. Ein Autor mehrere Bücher schreiben kann.
- Die Author-Tabelle enthält Informationen über die Autoren, darunter Id, FirstName, LastName, und Biography.
  - Die Books-Tabelle enthält Informationen über die Bücher, darunter Id, Title, ISBN, PublishedDate, BookType, und zusätzlich AuthorId als Fremdschlüssel, um den jeweiligen Autor des Buches zu referenzieren.

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

### Angabe: Ausleihsystem für Bücher (m..n)

In einem Bibliothekssystem können Kunden Bücher ausleihen. Jede Ausleihe umfasst Informationen darüber, welcher Kunde welches Buch an welchem Datum ausgeliehen hat, bis wann das Buch zurückgegeben werden soll, und welcher Bibliothekar die Ausleihe bearbeitet hat. Wenn ein Buch zurückgegeben wird, wird zusätzlich vermerkt, wann und welcher Bibliothekar die Rückgabe bearbeitet hat.

Erstelle ein Ausleihsystem mit einer m

zwischen Customer und Book, indem du eine Zwischentabelle BookLoan erstellst. Diese Tabelle speichert alle relevanten Informationen zur Ausleihe und Rückgabe.

- Die BookLoan-Tabelle soll die Ausleihen speichern und folgende Spalten umfassen:
  - CustomerId: Referenz auf den ausleihenden Kunden.
  - BookId: Referenz auf das ausgeliehene Buch.
  - LoanDate: Datum der Ausleihe.
  - DueDate: Datum, bis wann das Buch zurückgegeben werden soll.
  - LibrarianId: Bibliothekar, der die Ausleihe bearbeitet hat.
  - ReturnLibrarianId: Optionaler Bibliothekar, der die Rückgabe bearbeitet hat (nullable).
  - ReturnDate: Optionales Datum der Rückgabe (nullable).
- Jeder Kunde kann mehrere Bücher ausleihen, und jedes Buch kann mehrfach ausgeliehen werden.
- Erstelle einen passenden zusammengesetzten Primärschlüssel

```
-- Beispielhafte Ausleihe-Daten in die BookLoan-Tabelle

-- Kunde leiht ein Buch aus
INSERT INTO BookLoans (CustomerId, BookId, LoanDate, DueDate, LibrarianId)
VALUES (1, 1, '2024-01-10', '2024-02-10', 2);

-- Ein weiteres Beispiel mit Rückgabedaten
INSERT INTO BookLoans (CustomerId, BookId, LoanDate, DueDate, LibrarianId,
ReturnDate, ReturnLibrarianId)
VALUES (2, 3, '2024-02-01', '2024-02-28', 1, '2024-02-20', 3);

-- Noch eine Ausleihe ohne Rückgabedatum
INSERT INTO BookLoans (CustomerId, BookId, LoanDate, DueDate, LibrarianId)
VALUES (3, 2, '2024-02-15', '2024-03-15', 2);

-- Rückgabe eines anderen Buches
INSERT INTO BookLoans (CustomerId, BookId, LoanDate, DueDate, LibrarianId,
ReturnDate, ReturnLibrarianId)
VALUES (1, 4, '2024-03-05', '2024-04-05', 3, '2024-03-25', 1);
```