Dr. Holger Schwichtenberg

Moderne Datenzugriffslösungen mit Entity Framework Core 9.0

Datenbankprogrammierung / Objekt-Relationales Mapping mit C# in .NET 8.0 und .NET 9.0

www.IT-Visions.de®
Dr. Holger Schwichtenberg

Buchversion:	13. 0.2 (03.11.2024)
Verlag:	www.IT-Visions.de, Fahrenberg 40b, D-45257 Essen
Sprachliche Korrektur:	Dorotha Fleischer, Heike Rickert, Matthias Bloch (M.A.)
ISBN:	978-3-934-27928-5
Formate:	Druck, Kindle, PDF, PDF-Abo
Bezugsquellen:	*www.IT-Visions.de/Buch/EFC13*

Für Heidi, Felix und Maja

1 Inhaltsverzeichnis (Hauptkapitel)

2 Inhaltsverzeichnis (Details)

3 Vorwort

Liebe Leserinnen und Leser,

ich nutze Entity Framework in echten Softwareentwicklungsprojekten seit der allerersten Version, also seit der Version 1.0 von ADO.NET Entity Framework aus dem Jahr 2008. Zuvor hatte ich einen selbstentwickelten Objekt-Relationalen Mapper in meinen .NET-Projekten verwendet. Entity Framework Core ist das Nachfolgeprodukt, das es seit 2016 gibt. Ich setzte seitdem auch (aber nicht ausschließlich) Entity Framework Core in der Praxis ein. Einige meiner Projekte laufen noch mit dem klassischen Entity Framework.

Microsoft entwickelt Entity Framework Core inkrementell, d.h., die ersten Versionen waren noch sehr unvollständig. Mit jeder neuen Version schließt Microsoft weitere Lücken. Inzwischen hat Entity Framework Core eine gute Reife erreicht – wenngleich es immer noch einzelne Schwächen (wie in jeder Software) gibt, und diese thematisiert dieses Buch natürlich ebenfalls.

Genau so wie sich Entity Framework Core weiterentwickelt hat, so hat sich auch dieses Fachbuch seit der ersten Version im September 2016 kontinuierlich weiterentwickelt. Die vor Ihnen liegende Ausgabe beschreibt alle Kernaspekte und viele Tipps und Tricks sowie Praxisszenarien zu Entity Framework Core einschließlich der **Version 9.0**. Ich plane, in Zukunft weitere Versionen dieses Buchs zu veröffentlichen, die die kommenden Versionen von Entity Framework Core behandeln werden und auch weitere Tipps & Tricks sowie Praxisszenarien ergänzen.

Das Buch setzt aber voraus, dass Sie .NET, C# und Visual Studio schon kennen. Ich biete dazu aber auch weitere Bücher an, siehe Kapitel "Notwendige Vorkenntnisse".

Dieses Fachbuch wird vertrieben auf folgenden Wegen (ich nenne neben dem Verkaufspreis auch, wie viel – bzw. wenig – ich als Autor von den jeweiligen Händlern erhalte. Der Rest ist Gewinn der Händler):

- Gedruckt bei Amazon.de für 69,99 Euro (der Autor erhält 25,30 Euro):
  www.amazon.de/exec/obidos/ASIN/3934279295/itvisions-21
- Kindle-E-Book bei Amazon.de für 59,99 Euro (der Autor erhält 19,62 Euro):
  www.amazon.de/exec/obidos/ASIN/B0CMBZW766/itvisions-21
- PDF-E-Book bei Leanpub.com ab 49,99 Dollar (der Autor erhält 39,99 Dollar):
  www.leanpub.com/EntityFrameworkCore9
- Als Teil des E-Book-Buch-Abos zusammen mit anderen aktuellen Fachbüchern für 99,00 Euro im Jahr **inkl. aller Updates** (der Autor erhält den kompletten Preis):
  www.IT-Visions.de/BuchAbo

Tipp: Käufer bei Leanpub.com können jederzeit Aktualisierungen des PDF-Buchs (gleiche Hauptversion) kostenfrei dort beziehen. Käufer bei Amazon erhalten die PDF-Ausgabe einmalig kostenfrei (siehe Kapitel "Über dieses Fachbuch").

Da diese Preise in Anbetracht der vielen Stunden Arbeit an diesem Buch leider nicht nennenswert dazu beitragen können, den Lebensunterhalt meiner Familie zu bestreiten, ist dieses Projekt ein Hobby. Dementsprechend kann ich nicht garantieren, wann es Updates zu diesem Buch geben wird. Ich werde dann an diesem Buch arbeiten, wenn ich neben meinem Beruf als Softwarearchitekt, Berater und Dozent und meinen sportlichen Betätigungen noch etwas Zeit für das Fachbuchautorenhobby übrighabe.

Falls mir in diesem Buch oder den zugehörigen Downloads menschliche Fehler passiert sind, möchte ich mich dafür schon jetzt in aller Form bei Ihnen entschuldigen. Bitte geben Sie mir einen freundlichen, genau beschriebenen Hinweis auf meine Fehler. Ich freue mich immer über konstruktives Feedback und Verbesserungsvorschläge. Bitte verwenden Sie dazu das Kontaktformular: *www.dotnet-doktor.de/Leserfeedback*

Tipp: Ich belohne Sie mit E-Books für gemeldete Fehler, siehe Kapitel "Über dieses Buch/ Ihre Belohnung, wenn Sie helfen, dieses Buch zu verbessern".

Ich helfe Ihnen gerne dabei, Ihren eigenen Programmcode zu schreiben, aber ich hoffe, Sie verstehen, dass ich dies nicht ehrenamtlich tun kann. Wenn Sie **technische Hilfe** zu Entity Framework und Entity Framework Core oder anderen Themen rund um die Entwicklung und den Betrieb von Anwendungen (Desktop, Web und Mobile) sowie Server und Cloud benötigen, stehe ich Ihnen im Rahmen meiner

beruflichen Tätigkeit für die Firma *www.IT-Visions.de* (Beratung, Schulung, Support, Softwareentwicklung) gerne zur Verfügung. Bitte wenden Sie sich für ein Angebot an das jeweilige Kundenteam. Bitte kontaktieren Sie die Firmen aber nicht für Feedback und Verbesserungsvorschläge zu diesem Buch, da dieses Buch reine Privatsache ist.

Die Beispiele zu diesem Buch können Sie herunterladen auf der von mir ehrenamtlich betriebenen **Leser-Website** unter *www.dotnet-doktor.de/Leser*. Dort müssen Sie sich registrieren. Bei der Registrierung wird ein Losungswort abgefragt. Bitte geben Sie dort **Prodigy** ein (siehe auch Kapitel "Programmcodebeispiele zum Download").

Herzliche Grüße aus Essen, dem Herzen der Metropole Ruhrgebiet

Holger Schwichtenberg

4 Über den Autor

- Studienabschluss Diplom-Wirtschaftsinformatik an der Universität Essen
- Promotion an der Universität Essen im Fachgebiet komponentenbasierter Softwareentwicklung
- Seit 1996 in der IT tätig als Softwareentwickler, Softwarearchitekt, Berater, Dozent und Fachjournalist
- Fachlicher Leiter des Expertenteams bei *www.IT-Visions.de* in Essen
- Über 95 Fachbücher bei verschiedenen Verlagen, u.a. Carl Hanser Verlag, O'Reilly, APress, Microsoft Press und Addison Wesley sowie im Selbstverlag
- Mehr als 1500 Beiträge in Fachzeitschriften und Online-Portalen
- Gutachter in den Wettbewerbsverfahren der EU vs. Microsoft (2006-2009)
- Ständiger Mitarbeiter der Zeitschriften iX (seit 1999), dotnetpro (seit 2000) und Windows Developer (seit 2010) sowie beim Online-Portal heise.de (seit 2008)
- Regelmäßiger Sprecher auf nationalen und internationalen Fachkonferenzen (z.B. BASTA!, Developer Week, .NET Developer Conference, MD DevDays, Microsoft TechEd, Microsoft Summit, Microsoft IT Forum, OOP, .NET Architecture Camp, IT Tage, enterJS, Advanced Developers Conference, DOTNET Cologne, iterate=>ruhr, Community in Motion, DOTNET-Konferenz, VS One, NRW.Conf, Windows Forum, Container Conf)
- Auszeichnungen und Zertifikate von Microsoft:
  - Microsoft Most Valuable Professional (MVP), ununterbrochen ausgezeichnet seit 2004
  - Microsoft Certified Solution Developer (MCSD)
- Thematische Schwerpunkte:
  - Softwarearchitektur, mehrschichtige Softwareentwicklung, Softwarekomponenten
  - Visual Studio, Continous Integration (CI) und Continous Delivery (CD) mit Azure DevOps
  - Microsoft .NET (.NET Framework, .NET Core, modernes .NET), C#, Visual Basic
  - .NET-Architektur, Auswahl von .NET-Techniken
  - Einführung von .NET, Migration auf .NET
  - Webanwendungsentwicklung und Cross-Plattform-Anwendungen mit HTML/CSS, JavaScript/ TypeScript und C# sowie Webframeworks wie Angular, Vue.js, Svelte, ASP.NET (Core) und Blazor
  - Verteilte Systeme/Webservices mit .NET, insbesondere WebAPI, gRPC und WCF/CoreWCF
  - Relationale Datenbanken, XML, Datenzugriffsstrategien
  - Objekt-Relationales Mapping (ORM), insbesondere ADO.NET Entity Framework und Entity Framework Core
  - PowerShell
  - Architektur- und Code-Reviews
  - Performance-Analysen und -Optimierung
  - Entwicklungsrichtlinien

- Ehrenamtliche Community-Tätigkeiten:
  - Vortragender für die International .NET Association (INETA) und .NET Foundation
  - Betrieb diverser Community-Websites:
    www.dotnet-lexikon.de
    www.dotnetframework.de
    www.windows-scripting.de
    www.aspnetdev.de
    u.a.
- Firmenwebsite: *www.IT-Visions.de*
- Weblog: *www.dotnet-doktor.de*
- Kontakt für Anfragen zu Schulung und Beratung sowie Softwareentwicklungsprojekten:
  E-Mail *kundenteam@IT-Visions.de*
  Telefon *0201 / 64 95 90 – 50*

Kontakt für Feedback zu diesem Buch:
www.dotnet-doktor.de/Leserfeedback

5 Über dieses Buch

5.1 Versionsgeschichte dieses Buchs

Die Versionsgeschichte dieses Buch finden Sie in einem eigenen Kapitel am Ende des Buchs.

Hinweis: Die Versionsgeschichte ist eine wichtige Referenz für die Leser, die sich aktuelle Versionen des Buchs beschaffen (z.B. über Leanpub.com) und wissen wollen, was sich geändert hat. Wenn Sie das Buch erstmalig lesen, müssen Sie die Versionsgeschichte nicht lesen.

5.2 Hinweis zu den Vertriebswegen

Dieses Fachbuch wird vertrieben auf folgenden Wegen (Ich nenne neben dem Verkaufspreis auch, wie viel – bzw. wenig – ich als Autor von den jeweiligen Händlern erhalte. Der Rest ist Gewinn der Händler!):

- Gedruckt bei Amazon.de für 69,99 Euro (der Autor erhält 25,30 Euro):
  www.amazon.de/exec/obidos/ASIN/3934279295/itvisions-21
- Kindle-E-Book bei Amazon.de für 59,99 Euro (der Autor erhält 19,62 Euro):
  www.amazon.de/exec/obidos/ASIN/B0CMBZW766/itvisions-21
- PDF-E-Book bei Leanpub.com ab 49,99 Dollar (der Autor erhält 39,99 Dollar):
  www.leanpub.com/EntityFrameworkCore9
- Als Teil des E-Book-Buch-Abos zusammen mit anderen aktuellen Fachbüchern für 99,00 Euro im Jahr **inkl. aller Updates** (der Autor erhält den kompletten Preis):
  www.IT-Visions.de/BuchAbo

Hinweise: Ich habe mich für den Vertriebsweg des gedruckten Buchs über Amazon entschieden, weil ich dort ständig Updates zu dem Buch einreichen kann. Per Print-on-Demand erhalten Leser dann immer das topaktuelle Buch. Oft liefert Amazon dennoch am Tag nach der Bestellung das Buch schon aus. Der Vertrieb dieses Buch über klassische IT-Verlage, die leider heutzutage immer noch größere Auflagen vorproduzieren, ist für ein sehr agiles Softwareprodukt wie Entity Framework Core keine Alternative mehr.

Ich nenne dabei auch den Erlös für den Autor, weil ich sehr häufig Leser treffe, die fälschlicherweise denken, der wesentliche Teil des Buchpreises komme dem Autor zu Gute. Das ist leider nicht so, außer bei Leanpub.com oder eigenen Vertriebswegen wie meinem Buchabo. Daher denke ich, dass es sinnvoll ist, dies transparent zu machen.

Tipp: Käufer bei Leanpub.com können jederzeit Aktualisierungen des PDF-Buchs (gleiche Hauptversion) kostenfrei dort beziehen. Käufer bei Amazon erhalten die PDF-Ausgabe einmalig kostenfrei (siehe Kapitel "Bezugsquelle des PDF-E-Books"). E-Book-Abonnenten haben jederzeit Zugriff auf alle aktuellsten Ausgaben der Fachbücher von Dr. Holger Schwichtenberg.

5.3 Bezugsquelle des PDF-E-Books für Amazon-Kunden

Wenn Sie dieses Buch in gedruckter Form oder als Kindle-Ausgabe bei Amazon erworben haben, können Sie zusätzlich eine PDF-Version des Buchs **kostenfrei** erhalten.

Leiten Sie dazu Ihren Kaufbeleg von Amazon an folgende E-Mail-Adresse weiter:

PDFBuchZugabe@dotnet-doktor.de

Geben Sie dabei bitte Vorname, Name, Firma und E-Mail-Adresse an.

Sie erhalten dann binnen 1-2 Wochen das auf Sie personalisierte PDF-Dokument. Dieses Angebot gilt innerhalb von 6 Monaten nach dem Kauf des Buchs bei Amazon.

5.4 Bezugsquelle für Aktualisierungen

Sie können jederzeit Aktualisierungen des PDF-Buchs (gleiche Hauptversion!) kostenfrei bei Leanpub.com beziehen.

Käufer der Kindle- oder Druck-Version können die aktuelle PDF-Version zum Preis von 15,00 Dollar (zzgl. 7% Mehrwertsteuer) unter folgender Webadresse beziehen:

hwww.leanpub.com/EntityFrameworkCore9/Prodigy

Hinweise: Leider erlauben Amazon und andere Buchhändler aufgrund der Buchpreisbindungsgesetze in Deutschland den Autoren grundsätzlich nicht, dass Leser eine Aktualisierung im Kindle-Format oder in gedruckter Form vergünstigt erhalten.

Bitte beachten Sie auch, dass die ISBN-Regularien erfordern, dass man bei einer Titeländerung bei neuer Produktversion eine neue ISBN vergeben und damit auch ein neues Buchprojekt bei Amazon und Leanpub erstellt werden muss.

5.5 Hinweise zur Breite und Tiefe dieses Buchs – Sie haben Einfluss!

Ein Fachbuch, das ein riesengroßes Themengebiet wie Entity Framework Core behandelt, kann nicht jedes Teilgebiet und jeden Aspekt der Programmiersprache behandeln, zumindest nicht in gleicher Tiefe. Dann würde solch ein Fachbuch über eintausend Seiten, in einigen Fällen sogar mehrere tausend Seiten umfassen.

Ich denke, dass ich nach aktuellem Stand der Technik und meinem Wissenstand etwa 1.500 Seiten zu Entity Framework Core schreiben könnte. Würden Sie so ein dickes (und entsprechend teures) Buch kaufen und lesen wollen?

Wie jeder Fachautor lese auch ich immer wieder Kritik, dass ein Leser ein bestimmtes Thema nicht oder nicht in ausreichender Tiefe behandelt sei in dem Buch. Das ist aus der Sicht des einzelnen Lesers sicherlich gerechtfertigt, aber wie jeder Fachautor muss ich eben zwingend eine Auswahl der Themen treffen. Gerne dokumentiere ich hier, wie ich persönlich diese Auswahl für meine Bücher treffe:

- Ich behandele im Buch die Themen, die wir in unserer Firma selbst in der Praxis brauchen.
- Ich behandele zusätzlich die Themen, die unsere Kunden in Beratungsgesprächen behandelt haben möchten.
- Ich behandele Themen, die ich spannend finde.

Folglich sind die Themen, die ich im Buch nicht oder nur kurz behandele für uns und unsere Kunden nicht relevant bzw. so selbsterklärend, dass es keine Fragen dazu gibt.

Natürlich kann das für Sie anders sein. Sie können mir immer gerne schreiben, wenn Sie ein Thema im Buch behandelt haben möchten. Ich sammele diese Anregungen und wenn es mehrere Zuschriften zu einem Thema gibt, dann kommt das Thema weit oben auf die Prioritätenliste. Ich denke, das ist ein faires Verfahren.

5.6 Geplante Kapitel

Die Reihenfolge der für die folgenden Versionen geplanten Kapitel ist hier zunächst alphabetisch angeordnet und entspricht nicht der Reihenfolge, in der die Kapitel erscheinen werden.

- AddDbContextFactory() und AddPooledDbContextFactory() (seit Entity Framework Core 5.0)
- Anpassung der Code-Generierung beim Reverse Engineering mit Handlebars (Paket EntityFrameworkCore.Scaffolding.Handlebars)
- Auditing mit Entity Framework Plus
- Azure Cosmos DB (seit Entity Framework Core 3.0)

- Bulk Insert mit Entity Framework Extensions [*https://entityframework-extensions.net/*]
- Change Tracking mit Proxies (UseChangeTrackingProxies())
- Command Interceptors (seit Entity Framework Core 3.0)
- Connection Resiliency / EnableRetryOnFailure() (seit Entity Framework Core 1.1)
- Diagnostik mit Diagnostic Source / Diagnostic Listener (seit Entity Framework Core 3.0)
- Einfügen expliziten Primärschlüsselwerten trotz aktivierten Autowerten
- Event Counter
- Filtered Include für abgeleitete Typen (seit Entity Framework Core 2.1)
- ExcludeFromMigrations()
- Geo-Datentypen mit NetTopologySuite (seit Entity Framework Core 2.2)
- Group Join
- IDesignTimeDbContextFactory (seit Entity Framework Core 5.0)
- IModelCacheKeyFactory [*https://docs.microsoft.com/de-de/ef/core/modeling/dynamic-model*]
- MonoDB (Treiber verfügbar seit Mai 2024)
- Parameter in Konstruktoren von Entitätsklassen (seit Entity Framework Core 2.1)
- Performance: ToArray() vs. ToList() vs. ToHashTable() vs. ToDictionary()
- Performance: Asynchrone Methoden vs. Synchrone Methoden
- Pre-Convention Model Configuration (seit Entity Framework Core 6.0)
- Projektstrukturen anlegen mit der .NET Core CLI
- Record-Typen (record class und record struct) mit Entity Framework Core
- Reverse Engineering von Oracle-Datenbanken mit Entity Framework Core Power Tools (seit August 2020 möglich)
- Savepoints in Transaktionen (seit Entity Framework Core 5.0)
- SaveChangesInterceptor (seit Entity Framework Core 5.0)
- Timestamps mit SQLite
- SQL Server Memory-Optimized Tables (seit Entity Framework Core 1.1)
- SQLite in Blazor WebAssembly-Anwendungen nutzen
- Trackable Entities für EF Core [*https://github.com/TrackableEntities/TrackableEntities.Core*]
- Transaction-IDs (seit Entity Framework Core 5.0)
- User Defined Functions (UDF)
- Value Conversion mit komplexen Typen (Mapping mehrerer Properties auf eine Spalte)
- Visual Studio Analyzer für SQL-Abfragen (seit Entity Framework Core 2.1)
- Zusätzliche Erweiterungen aus der Community wie z.B. EntityFrameworkCore.Rx, EFDetached.EntityFramework, EntityFrameworkCore.Triggers, EntityFrameworkCore.PrimaryKey und EntityFrameworkCore.TypedOriginalValues
- Zustandsänderungsereignisse im Change Tracker (seit Entity Framework Core 2.1)

Hinweis: Mit dieser Liste kommender Themen möchte ich gleichzeitig klarstellen, welche Themen Sie derzeit nicht im Buch finden, damit Sie als Leser keine falschen Erwartungen haben.

5.7 Programmiersprache in diesem Buch

Als Programmiersprache kommt in diesem Buch C# zum Einsatz, weil dies die bei weitem häufigste verwendete Programmiersprache in .NET ist. Der Autor dieses Buchs programmiert in einigen Kundenprojekten .NET-Anwendungen zwar auch in Visual Basic .NET, leider bietet dieses Buch jedoch nicht den Raum, alle Listings in beiden Sprachen wiederzugeben.

Eine Sprachkonvertierung zwischen C# und Visual Basic .NET ist im WWW kostenfrei verfügbar z.B. auf der Website *http://converter.telerik.com*

Falls Sie auf C# umsteigen möchten oder noch eine ältere Version von C# verwenden, möchte ich Ihnen mein Fachbuch "C# Crashkurs" empfehlen: *www.IT-Visions.de/CSharpBuch*

5.8 Notwendige Vorkenntnisse

Datenbankprogrammierung mit OR-Mappern ist ein großes, komplexes Gebiet. In diesem Buch konzentriere ich mich auf Entity Framework Core im engeren Sinne.

Als Vorkenntnisse sollten Sie zumindest ein gutes Grundlagenwissen in folgenden Gebieten mitbringen:

- Relationale Datenbanken und Datenbankdesign
- Structured Query Language (SQL)
- .NET
- C#
- Visual Studio

5.9 Hinweis auf einen Bug im PDF-Exporter von Word

Nutzer der PDF-Version dieses Buch werden feststellen, dass die Hyperlinks an einigen Stellen im Buch nicht zu funktionieren scheinen. Ich möchte Ihnen aber versichern, dass

- ich die Hyperlinks per Copy&Paste aus dem Browser ohne Verlust in mein Manuskript in Microsoft Word übernommen habe und
- die Links innerhalb des Word-Dokuments funktionieren.

Allerdings hat der PDF-Exporter von Word einen seit langem bekannten und von Microsoft immer noch nicht behobenen Bug: Der PDF-Exporter nimmt manchmal einen Buchstaben zu viel in den Link, z.B. aus diesem Hyperlink in Word

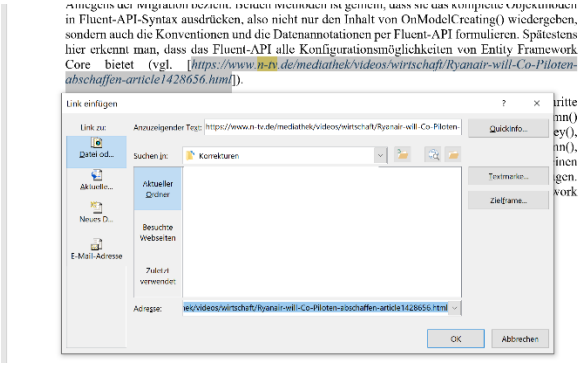

Abbildung: Korrekter Link im Word-Dokument

wird dann im PDF ein Link, bei dem die schließende Klammer fehlerhafter Weise mit aufgenommen wird.

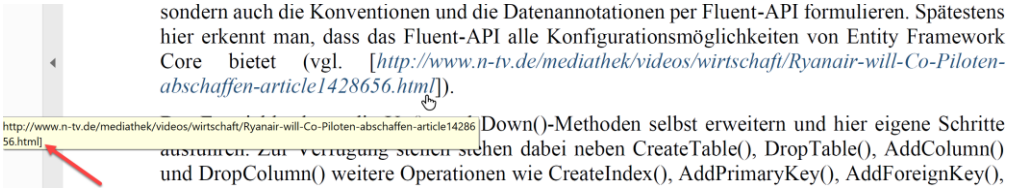

Abbildung: Fehlerhafter Link mit einem Buchstaben zu viel im PDF-Dokument, das Word erzeugt

Ich bedauere das Problem, bitte Sie allerdings, sich in dieser Sache an Microsoft zu wenden, da ich das nicht ändern kann. Ich kann nur einen Tipp zur Problemumgehung geben: Hyperlink per Copy & Paste rauskopieren in den Browser.

5.10 Ihre Belohnung, wenn Sie helfen, dieses Buch zu verbessern!

Wenn Sie Fehler in diesem Buch finden, bin ich Ihnen nicht nur wirklich sehr dankbar, sondern Sie bekommen auch eine Belohnung in Form von aktualisierten oder weiteren E-Books.

Fehlerart	E-Book-Guthaben
Inhaltlicher Fehler	Pro Fehler 5 Euro
Sprachlicher Fehler	Pro Fehler 2 Euro

Ein Beispiel: Wenn Sie zwei inhaltliche Fehler und zehn Rechtschreibfehler in diesem Buch finden, dann haben Sie bei mir 30 Euro gut. Dafür können Sie dann eins meiner selbstverlegten Bücher als E-Book bekommen.

Die selbstverlegten Bücher finden Sie unter *www.IT-Visions.de/Verlag*

Melden Sie die Fehler unter *www.dotnet-doktor.de/Leserfeedback*

Schreiben Sie dabei, welches E-Book Sie wünschen. Das Buch schicke ich Ihnen dann per E-Mail zu.

Tipp: Auch Fehler auf meiner persönlichen Website *www.dotnet-doktor.de* und der Firmenwebsite *www.IT-Visions.de* zählen mit!

Ich freue mich auf Ihre Fehlermeldung!

Holger Schwichtenberg

P.S. Die Fehlermeldung zählt nur, wenn nicht ein anderer Leser dies bereits gemeldet hat und es daher in der aktuellen Auflage schon korrigiert ist.

6 Fallbeispiele in diesem Buch

Die meisten Beispielprogrammcodes in diesem Buch drehen sich um das Fallbeispiel der fiktiven Fluggesellschaft "WorldWideWings", abgekürzt "WWWings" oder als dreibuchstabiger Airline Code einfach "WWW". Es gibt auch eine Website zu der Fluggesellschaft (*www.world-wide-wings.de*) – dort einen Flug zu buchen, möchte der Autor dieses Buchs Ihnen aber nicht empfehlen ☺

Abbildung: Logo der fiktiven Fluggesellschaft "WorldWideWings"

Hinweis: In einzelnen Unterkapitel werden andere Fallbeispiele verwendet (z.B. die Aufgabenverwaltung "MiracleList"). Diese Fallbeispiele werden dann in den jeweiligen Kapiteln erläutert.

6.1 Entitäten

Im Anwendungsfall "WorldWideWings" geht es um folgende Entitäten:

- **Flüge** zwischen zwei Orten, bei denen die Orte bewusst nicht als eigene Entität modelliert wurden, sondern Zeichenketten sind (dies vereinfacht das Verständnis vieler Beispiele).
- **Passagiere**, die auf Flügen fliegen.
- **Mitarbeiter** der Fluggesellschaft, die wiederum Vorgesetzte haben, die auch Mitarbeiter sind.
- **Piloten** als eine Spezialisierung von Mitarbeitern. Ein Flug hat im einfacheren Modell nur einen Piloten. Es gibt keinen Copiloten bei WorldWideWings. Den Copiloten abzuschaffen und im Notfall das Flugzeug von der Stewardess landen zu lassen (wie im Film "Turbulence" von 1997) war übrigens im Jahr 2010 ein echter Vorschlag von Michael O'Leary, dem Chef der irischen Fluggesellschaft Ryanair (siehe [vgl. [*http://www.n-tv.de/mediathek/videos/wirtschaft/Ryanair-will-Co-Piloten-abschaffen-article1428656.html*]] und [*http://www.dailymail.co.uk/news/article-1308852/Let-stewardesses-land-plane-crisis-says-Ryanair-boss-Airline-wants-ditch-pilots.html*]).
- **Personen** als Sammlung der gemeinsamen Eigenschaften für alle Menschen in diesem Beispiel. Personen gibt es aber nicht eigenständig, sondern nur in den Ausprägungen/Spezialisierungen Passagier, Mitarbeiter und Pilot. Im objektorientierten Sinne ist Person also eine abstrakte Basisklasse, die keine Instanzen besitzen kann, sondern nur der Vererbung dient.

Es gibt zwei Datenmodelle:

- Das etwas einfachere Modell #1 (alias Modell Version 1, siehe Abbildungen 1 und 2) ist das Ergebnis klassischen relationalen Datenbankdesigns mit Normalisierung. Das Objektmodell daraus entsteht per Reverse Engineering.
- Modell #2 (alias Modell Version 2, siehe Abbildungen 3 und 4) ist das Ergebnis des Forward Engineering mit Entity Framework Core aus einem Objektmodell. Zusätzlich gibt es hier weitere Entitäten (Persondetail, Flugzeugtyp und Flugzeugtypdetail), um weitere Modellierungsaspekte aufzeigen zu können. In diesem Fall gibt es auch für jeden Flug einen optionalen Copiloten.

In Modell #1 gibt es eine jeweils eigene Tabelle für Personen (auch wenn es keine eigenständigen Personen gibt), Mitarbeiter, Piloten und Passagiere. Diese Aufteilung entspricht den Klassen im Objektmodell.

Hinweise: Das Objektmodell zum Datenbankschema WorldWideWings Version 1 (Abbildung 2) ist das automatisch von Entity Framework Core aus der Datenbank generierte Objektmodell (Reverse Engineering); es ist bewusst nicht verändert worden, auch wenn einige der generierten Namen unschön sind.

Die Möglichkeit der N:M-Abstraktion, die es in Entity Framework Core seit Version 5.0 gibt, ist in dem Objektmodell noch nicht umgesetzt, es beim Reverse Engineering in Version 5.0 noch keine N:M-Abstraktion gibt; dies soll erst in Version 6.0 kommen.

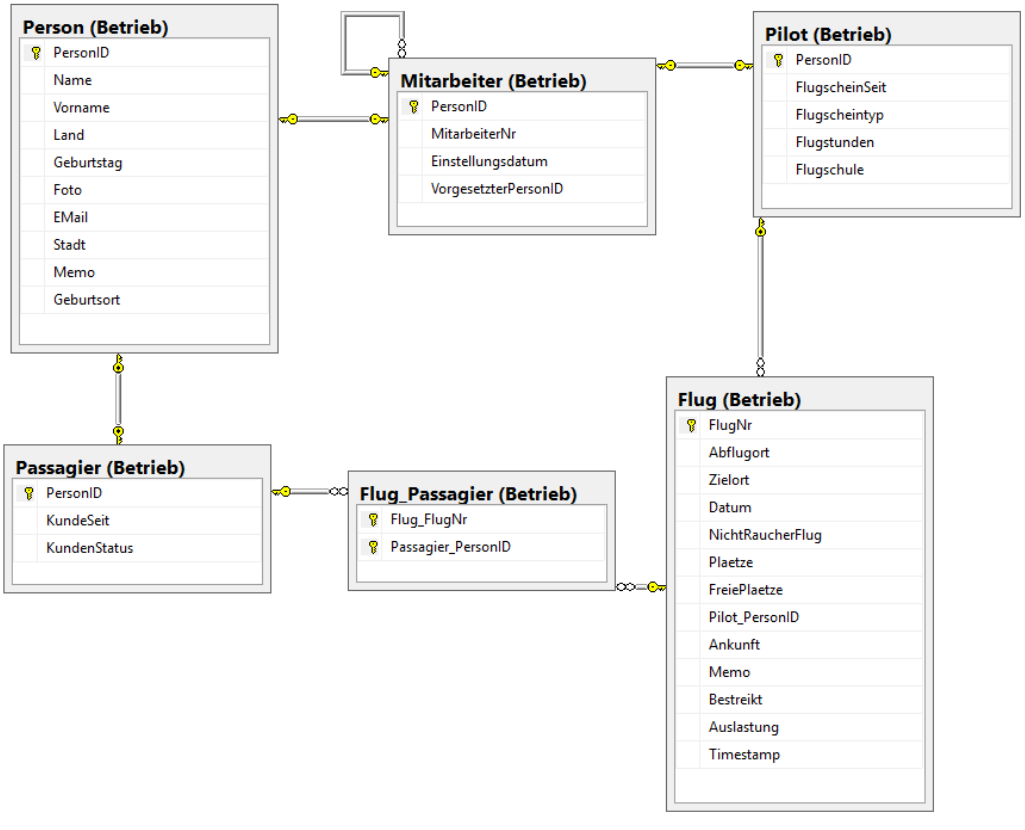

Abbildung 1: WorldWideWings-Datenmodell in der einfacheren Version 1

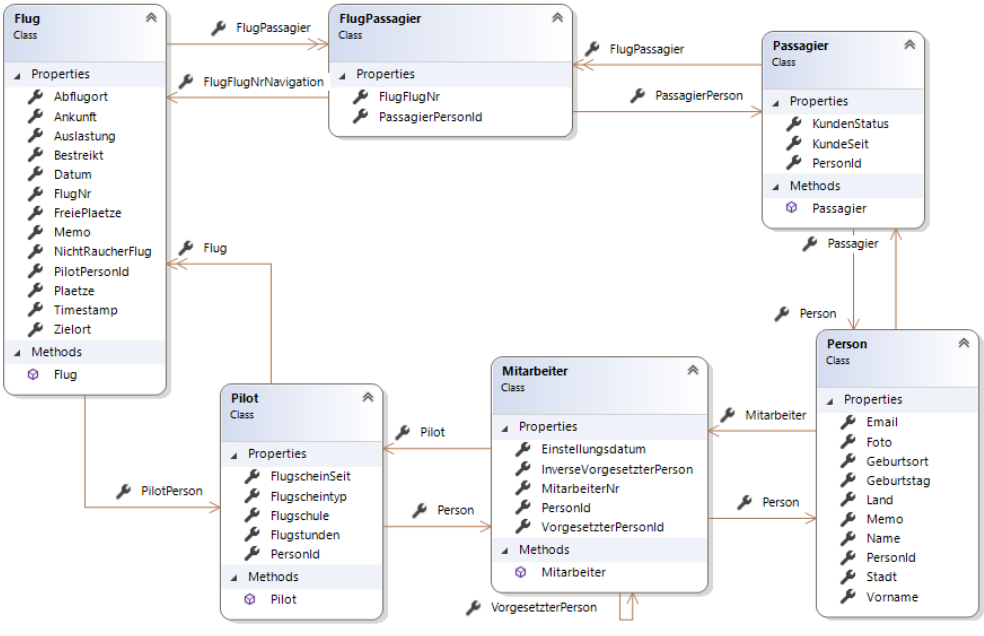

Abbildung 2: Objektmodell zum WorldWideWings-Datenmodell in der einfacheren Version 1

In Modell #2 gibt es lediglich die Tabellen Passagiere und Mitarbeiter für diese vier Entitäten. Entity Framework Core ist derzeit etwas eingeschränkt und unterstützt das "Table-per-Type"-Mapping (also eine eigenständige Tabelle für jede Klasse) nicht. Daher umfasst die Tabelle Passagiere auch alle Eigenschaften von Person. Die Tabelle Mitarbeiter umfasst neben den Personeneigenschaften die Eigenschaften der Entitäten Mitarbeiter und Pilot. In der Tabelle wird per Diskriminatorspalte unterschieden zwischen Datensätzen, die ein Mitarbeiter sind, und solchen, die ein Pilot sind. Entity Framework Core mischt hier die Konzepte Table-per-Concrete-Type (TPC) und Table-per-Hierarchy (TPH). Einen dezidierten Einfluss auf diese Abbildung hat man in Entity Framework Core 1.x/2.0 noch nicht. Das klassische Entity Framework bietet hier mehr Optionen.

Die Abhängigkeitsarten in Modell #2 sind:

- Ein **Flug** muss einen Piloten besitzen. Es gibt einen Copiloten, aber er ist optional.
- Ein Flug kann optional einen **Flugzeugtyp** zugeordnet haben. Ein Flugzeugtyp hat eine Beziehung zu **Flugzeugtypdetail**.
- Jede Person und damit auch jeder Pilot und Passagier muss ein **Persondetail**-Objekt besitzen.

In diesem Buch kommen beide Datenmodelle vor, teilweise auch in modifizierter Form, um bestimmte Szenarien (z.B. Datenbankschemamigrationen) aufzuzeigen.