

Inhalt

Vorwort	XV
Über den Autor	XVII
1 Einleitung	1
1.1 Programmiersprache in diesem Buch	1
1.2 Fallbeispiele in diesem Buch	1
1.2.1 Entitäten	2
1.3 Anwendungsarten in diesem Buch	5
1.4 Hilfsroutinen zur Konsolenausgabe	6
1.5 Programmcodebeispiel zum Download	10
2 Was ist Entity Framework Core?	13
2.1 Was ist ein Objekt-Relationaler Mapper?	13
2.2 ORM in der .NET-Welt	15
2.3 Versionsgeschichte von Entity Framework Core	16
2.4 Unterstützte Betriebssysteme	17
2.5 Unterstützte .NET-Versionen	17
2.6 Unterstützte Visual Studio-Versionen	18
2.7 Unterstützte Datenbanken	19
2.8 Funktionsumfang von Entity Framework Core	20
2.9 Funktionen, die dauerhaft entfallen	21
2.10 Funktionen, die Microsoft bald nachrüsten will	21
2.11 Hohe Priorität, aber nicht kritisch	22
2.12 Neue Funktionen in Entity Framework Core	23
2.13 Einsatzszenarien	24
3 Installation von Entity Framework Core	27
3.1 Nuget-Pakete	27

3.2	Paketinstallation	30
3.3	Aktualisierung auf eine neue Version	34
4	Konzepte von Entity Framework Core	39
4.1	Vorgehensmodelle bei Entity Framework Core	39
4.2	Artefakte bei Entity Framework Core	41
5	Reverse Engineering bestehender Datenbanken	43
5.1	Reverse Engineering mit PowerShell-Befehlen	44
5.2	Codegenerierung	47
5.3	.NET Core-Tool	54
5.4	Schwächen des Reverse Engineering	56
6	Forward Engineering für neue Datenbanken	57
6.1	Regeln für die selbsterstellten Entitätsklassen	59
6.1.1	Properties	59
6.1.2	Datentypen	59
6.1.3	Beziehungen (Master-Detail)	59
6.1.4	Vererbung	61
6.1.5	Primärschlüssel	61
6.1.6	Beispiele	61
6.2	Regeln für die selbsterstellte Kontextklasse	64
6.2.1	Nuget-Pakete	64
6.2.2	Basisklasse	65
6.2.3	Konstruktor	65
6.2.4	Beispiel	65
6.2.5	Provider und Verbindungszeichenfolge	66
6.2.6	Eigene Verbindungen	67
6.2.7	Thread-Sicherheit	67
6.3	Regeln für die Datenbankschemagenerierung	67
6.4	Beispiel-Client	68
6.5	Anpassung per Fluent-API (OnModelCreating())	69
6.6	Das erzeugte Datenmodell	71
7	Anpassung des Datenbankschemas	73
7.1	Persistente versus transiente Klassen	74
7.2	Namen im Datenbankschema	75
7.3	Reihenfolge der Spalten in einer Tabelle	75
7.4	Spaltentypen/Datentypen	76
7.5	Pflichtfelder und optionale Felder	77
7.6	Feldlängen	77
7.7	Primärschlüssel	78

7.8 Beziehungen und Fremdschlüssel	78
7.9 Optionale Beziehungen und Pflichtbeziehungen	79
7.10 Uni- und Bidirektionale Beziehungen	81
7.11 1:1-Beziehungen	82
7.12 Indexe festlegen	83
7.13 Weitere Syntaxoptionen für das Fluent-API	85
7.13.1 Sequentielle Konfiguration	85
7.13.2 Strukturierung durch Statement Lambdas	85
7.13.3 Strukturierung durch Unterroutinen	86
7.13.4 Strukturierung durch Konfigurationsklassen	87
7.14 Massenkfiguration mit dem Fluent-API	88
8 Datenbankschemamigrationen	91
8.1 Anlegen der Datenbank zur Laufzeit	91
8.2 Schemamigrationen zur Entwicklungszeit	92
8.3 Befehle für die Schemamigrationen	92
8.4 ef.exe	93
8.5 Add-Migration	94
8.6 Update-Database	98
8.7 Script-Migration	99
8.8 Weitere Migrationsschritte	99
8.9 Migrationsszenarien	100
8.10 Weitere Möglichkeiten	101
8.11 Schemamigrationen zur Laufzeit	103
9 Daten lesen mit LINQ	105
9.1 Kontextklasse	105
9.2 LINQ-Abfragen	106
9.3 Schrittweises Zusammensetzung von LINQ-Abfragen	108
9.4 Repository-Pattern	109
9.5 Einsatz von var	110
9.6 LINQ-Abfragen mit Paging	111
9.7 Projektionen	113
9.8 Abfrage nach Einzelobjekten	114
9.9 Laden anhand des Primärschlüssels mit Find()	115
9.10 LINQ im RAM statt in der Datenbank	115
9.11 Umgehung für das GroupBy-Problem	119
9.11.1 Mapping auf Nicht-Entitätstypen	119
9.11.2 Entitätsklasse für die Datenbanksicht anlegen	120
9.11.3 Einbinden der Entitätsklasse in die Kontextklasse	120
9.11.4 Verwendung der Pseudo-Entitätsklasse	121

9.11.5	Herausforderung: Migrationen	121
9.11.6	Gruppierungen mit Datenbanksichten	123
9.12	Kurzübersicht über die LINQ-Syntax	123
9.12.1	Einfache SELECT-Befehle (Alle Datensätze)	125
9.12.2	Bedingungen (where)	125
9.12.3	Bedingungen mit Mengen (in)	126
9.12.4	Sortierungen (orderby)	126
9.12.5	Paging (Skip() und Take())	127
9.12.6	Projektion	127
9.12.7	Aggregatfunktionen (Count(), Min(), Max(), Average(), Sum()) ..	128
9.12.8	Gruppierungen (GroupBy)	129
9.12.9	Einzelobjekte (SingleOrDefault(), FirstOrDefault())	130
9.12.10	Verbundene Objekte (Include())	131
9.12.11	Inner Join (Join)	132
9.12.12	Cross Join (Kartesisches Produkt)	133
9.12.13	Join mit Gruppierung	133
9.12.14	Unter-Abfragen (Sub-Select)	134
10	Objektbeziehungen und Ladestrategien	137
10.1	Standardverhalten	137
10.2	Kein Lazy Loading	138
10.3	Eager Loading	140
10.4	Explizites Nachladen (Explicit Loading)	143
10.5	Preloading mit Relationship Fixup	145
10.6	Details zum Relationship Fixup	150
11	Einfügen, Löschen und Ändern	151
11.1	Speichern mit SaveChanges()	151
11.2	Änderungsverfolgung auch für Unterobjekte	154
11.3	Das Foreach-Problem	155
11.4	Objekte hinzufügen mit Add()	156
11.5	Verbundene Objekte anlegen	158
11.6	Verbundene Objekte ändern/Relationship Fixup	161
11.7	Widersprüchliche Beziehungen	164
11.8	Zusammenfassen von Befehlen (Batching)	169
11.9	Objekte löschen mit Remove()	170
11.10	Löschen mit einem Attrappen-Objekt	171
11.11	Massenlöschen	172
11.12	Datenbanktransaktionen	173
11.13	Change Tracker abfragen	176
11.13.1	Zustand eines Objekts	176
11.13.2	Liste aller geänderten Objekte	178

12	Datenänderungskonflikte (Concurrency)	181
12.1	Rückblick	181
12.2	Im Standard keine Konflikterkennung	182
12.3	Optimistisches Sperren/Konflikterkennung	184
12.4	Konflikterkennung für alle Eigenschaften	185
12.5	Konflikteinstellung per Konvention	186
12.6	Fallweise Konflikteinstellung	187
12.7	Zeitstempel (Timestamp)	188
12.8	Konflikte auflösen	190
12.9	Pessimistisches Sperren bei Entity Framework Core	194
13	Protokollierung (Logging)	199
13.1	Verwendung der Erweiterungsmethode Log()	199
13.2	Implementierung der Log()-Erweiterungsmethode	201
13.3	Protokollierungskategorien	204
14	Asynchrone Programmierung	205
14.1	Asynchrone Erweiterungsmethoden	205
14.2	ToListAsync()	205
14.3	SaveChangesAsync()	207
14.4	ForeachAsync()	208
15	Dynamische LINQ-Abfragen	211
15.1	Schrittweises Zusammensetzen von LINQ-Abfragen	211
15.2	Expression Trees	213
15.3	Dynamic LINQ	216
16	Daten lesen und ändern mit SQL, Stored Procedures und Table Valued Functions	219
16.1	Abfragen mit FromSql()	219
16.2	Zusammensetzbarkeit von LINQ und SQL	221
16.3	Globale Abfragefilter bei SQL-Abfragen (ab Version 2.0)	223
16.4	Stored Procedures und Table Valued Functions	224
16.5	Globale Abfragefilter bei Stored Procedures und Table Valued Functions	225
16.6	Nicht-Entitätsklassen als Ergebnismenge	226
16.7	SQL-DML-Befehle ohne Resultset	228
17	Weitere Tipps und Tricks zum Mapping	229
17.1	Shadow Properties	229
17.1.1	Automatische Shadow Properties	229
17.1.2	Festlegung eines Shadow Property	230

17.1.3	Ausgabe aller Shadow Properties einer Entitätsklasse	230
17.1.4	Lesen und Ändern eines Shadow Property	231
17.1.5	LINQ-Abfragen mit Shadow Properties	232
17.1.6	Praxisbeispiel: Automatisches Setzen des Shadow Property bei jedem Speichern	232
17.2	Tabellenaufteilung (Table Splitting)	233
17.3	Berechnete Spalten (Computed Columns)	236
17.3.1	Automatisches SELECT	237
17.3.2	Praxistipp: Spalten mit einer Berechnungsformel anlegen	237
17.3.3	Spalten mit einer Berechnungsformel nutzen	239
17.3.4	Spalten mit einer Berechnungsformel beim Reverse Engineering	241
17.4	Standardwerte (Default Values)	242
17.4.1	Standardwerte beim Forward Engineering festlegen	242
17.4.2	Standardwerte verwenden	243
17.4.3	Praxistipp: Standardwerte schon beim Anlegen des Objekts vergeben	245
17.4.4	Standardwerte beim Reverse Engineering	246
17.5	Sequenzobjekte (Sequences)	247
17.5.1	Erstellen von Sequenzen beim Forward Engineering	247
17.5.2	Sequenzen im Einsatz	248
17.6	Alternative Schlüssel	251
17.6.1	Alternative Schlüssel definieren	252
17.6.2	Alternative Schlüssel im Einsatz	254
17.7	Kaskadierendes Löschen (Cascading Delete)	255
17.8	Abbildung von Datenbanksichten (Views)	258
17.8.1	Datenbanksicht anlegen	259
17.8.2	Entitätsklasse für die Datenbanksicht anlegen	259
17.8.3	Einbinden der Entitätsklasse in die Kontextklasse	259
17.8.4	Verwendung der Datenbanksicht	260
17.8.5	Herausforderung: Migrationen	260
18	Weitere Tipps und Tricks zu LINQ	263
18.1	Globale Abfragefilter (ab Version 2.0)	263
18.1.1	Filter definieren	263
18.1.2	Filter nutzen	264
18.1.3	Praxistipp: Filter ignorieren	265
18.2	Zukünftige Abfragen (Future Queries)	265
19	Leistungsoptimierung (Performance Tuning)	267
19.1	Vorgehensmodell zur Leistungsoptimierung bei Entity Framework Core	267
19.2	Best Practices für Ihre eigenen Leistungstests	268
19.3	Leistungsvergleich verschiedener Datenzugriffstechniken in .NET	268
19.4	Objektzuweisung optimieren	270

19.5	Massenoperationen	272
19.5.1	Einzellöschen	272
19.5.2	Optimierung durch Batching	273
19.5.3	Löschen ohne Laden mit Pseudo-Objekten	274
19.5.4	Einsatz von klassischem SQL anstelle des Entity Framework Core-APIs	275
19.5.5	Lambda-Ausdrücke für Massenlöschen mit EFPlus	277
19.5.6	Massenaktualisierung mit EFPlus	278
19.6	Leistungsoptimierung durch No-Tracking	279
19.6.1	No-Tracking aktivieren	280
19.6.2	No-Tracking fast immer möglich	282
19.6.3	No-Tracking im änderbaren Datagrid	285
19.6.4	QueryTrackingBehavior und AsTracking()	286
19.6.5	Best Practices	288
19.7	Auswahl der besten Ladestrategie	288
19.8	Zwischenspeicherung (Caching)	289
19.8.1	MemoryCache	289
19.8.2	Abstraktion CacheManager	292
19.9	Second-Level-Caching mit EFPlus	296
19.9.1	Einrichten des Second-Level-Cache	297
19.9.2	Verwenden des Second-Level-Cache	297
20	Softwarearchitektur mit Entity Framework Core	301
20.1	Monolithisches Modell	301
20.2	Entity Framework Core als Datenzugriffsschicht	302
20.3	Reine Geschäftslogik	304
20.4	Geschäftsobjekt- und ViewModel-Klassen	305
20.5	Verteilte Systeme	306
20.6	Fazit	309
21	Zusatzwerkzeuge	311
21.1	Entity Framework Core Power Tools	311
21.1.1	Funktionsüberblick	311
21.1.2	Reverse Engineering mit Entity Framework Core Power Tools ...	312
21.1.3	Diagramme mit Entity Framework Core Power Tools	316
21.2	LINQPad	317
21.2.1	Aufbau von LINQPad	318
21.2.2	Datenquellen einbinden	319
21.2.3	LINQ-Befehle ausführen	322
21.2.4	Speichern	325
21.2.5	Weitere LINQPad-Treiber	325
21.2.6	Interaktive Programmcodeeingabe	326
21.2.7	Fazit zu LINQPad	327

21.3	Entity Developer	327
21.3.1	Reverse Engineering mit Entity Developer	328
21.3.2	Forward Engineering mit Entity Developer	339
21.4	Entity Framework Profiler	344
21.4.1	Einbinden des Entity Framework Profilers	346
21.4.2	Befehle überwachen mit Entity Framework Profiler	346
21.4.3	Warnungen vor potenziellen Problemen	349
21.4.4	Analysefunktionen	349
21.4.5	Fazit zu Entity Framework Profiler	350
22	Zusatzkomponenten	351
22.1	Entity Framework Plus (EFPlus)	351
22.2	Second-Level-Caching mit EFSecondLevelCache.Core	352
22.3	Objekt-Objekt-Mapping und AutoMapper	352
22.3.1	Objekt-Objekt-Mapping per Reflection	354
22.3.2	AutoMapper	357
22.3.3	Beispiel	358
22.3.4	Abbildungen konfigurieren	359
22.3.5	Abbildung ausführen mit Map()	360
22.3.6	Abbildungskonventionen	361
22.3.7	Profilklassen	363
22.3.8	Verbundene Objekte	363
22.3.9	Manuelle Abbildungen	364
22.3.10	Typkonvertierungen	366
22.3.11	Objektmengen	368
22.3.12	Vererbung	369
22.3.13	Generische Klassen	371
22.3.14	Zusatzaktionen vor und nach dem Mapping	374
22.3.15	Geschwindigkeit	375
22.3.16	Fazit zu AutoMapper	376
23	Praxislösungen	379
23.1	Entity Framework Core in einer ASP.NET Core-Anwendung	379
23.1.1	Das Fallbeispiel "MiracleList"	379
23.1.2	Architektur	383
23.1.3	Entitätsklassen	385
23.1.4	Entity Framework Core-Kontextklasse	387
23.1.5	Lebensdauer der Kontextklasse in ASP.NET Core-Anwendungen ..	389
23.1.6	Geschäftslogik	390
23.1.7	WebAPI	398
23.1.8	Verwendung von Entity Framework Core per Dependency Injection	408
23.1.9	Praxistipp: Kontextinstanzpooling (DbContext Pooling)	411
23.2	Entity Framework Core in einer Universal Windows Platform App	412

23.2.1	Das Fallbeispiel "MiracleList Light"	412
23.2.2	Architektur	413
23.2.3	Entitätsklassen	415
23.2.4	Entity Framework Core-Kontextklasse	416
23.2.5	Startcode	416
23.2.6	Erzeugte Datenbank	417
23.2.7	Datenzugriffscode	419
23.2.8	Benutzeroberfläche	423
23.3	Entity Framework Core in einer Xamarin-Cross-Platform-App	424
23.3.1	Das Fallbeispiel "MiracleList Light"	424
23.3.2	Architektur	426
23.3.3	Entitätsklassen	428
23.3.4	Entity Framework Core-Kontextklasse	428
23.3.5	Startcode	430
23.3.6	Erzeugte Datenbank	430
23.3.7	Datenzugriffscode	430
23.3.8	Benutzeroberfläche	434
23.4	N:M-Beziehungen zu sich selbst	435
24	Quellen im Internet	441
	Index	443