Entity Framework Core

Migrationen sind ein entscheidender Teil von Entity Framework Core.
Prinzipiell werden mit mithilfe von Migrationen die erstellten C# POCOs und der DbContext in eine Datenbank überführt.

Dabei verwandelt Entity Framework den Source Code mit den folgenden Mitteln in die Datenbank:

- C# Pocos
- DbContext
- Dotnet ef CLI tool
- Migrationsdateien und Snapshot
- Provider der Datenbank

Migrationen sind demnach abhängig vom Datenbank und vom Entityframework Provider der verwendet wird.

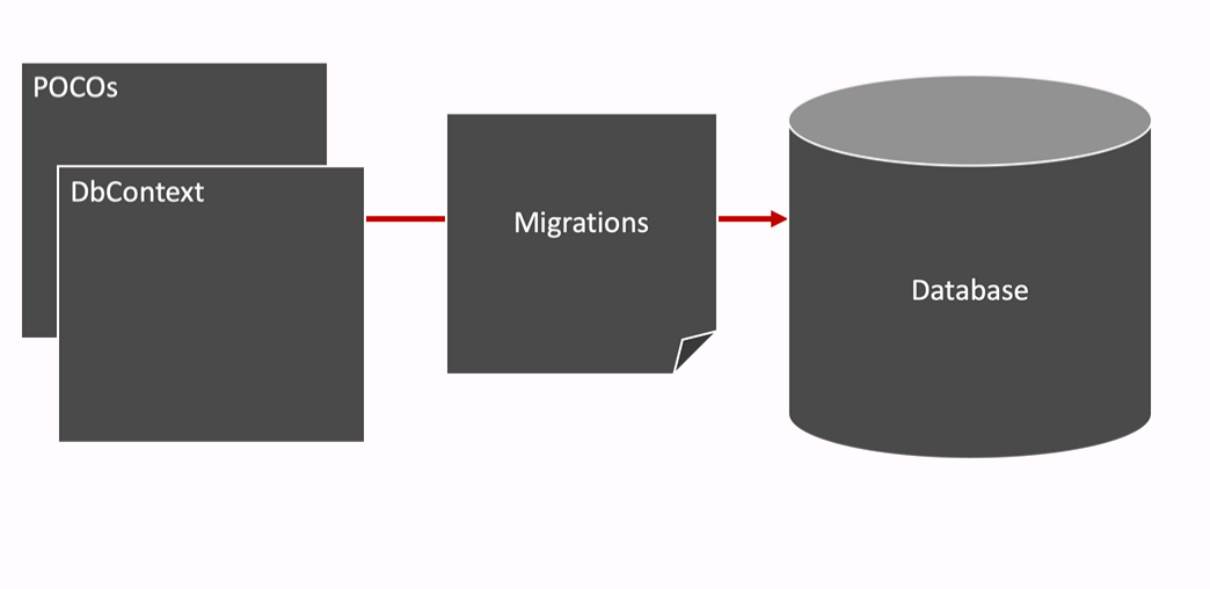

Abbildung 1 - Schematische Ansicht von Migrationen

Das ist aber noch nicht alles, Migrationen werden auch für jede Schemaänderung der Datenbank herangezogen.

Mit dem CLI Tool werden die Migrationsdateien für gewöhnlich automatisch generiert.
Manchmal kann es jedoch erforderlich sein, die Migrationsdateien anzupassen:

- Wenn der Provider die gewünschte Änderung nicht unterstützt
- Wenn Standardwerte eingetragen werden sollen
- Wenn selbst geschriebenes SQL performanter ist als die Migration
- Etc.

Migrationen anwenden

Migrationen an sich erstellen/verändern die vorhandene Datenbank noch nicht, sondern sind nur eine Vorlage.

Um die Datenbank endgültig anzupassen, benötigen wir eine Anwendung der Migration. Dafür gibt es 3 Möglichkeiten:

1. **Per SQL Skript**
   Wir können aus den erstellten Migrationen ein SQL Skript erzeugen. Dieses kann dann auf dem normalen Wege auf der Datenbank angewendet werden.
2. **Per CLI Tool**
   Der üblichste Weg ist wohl der über das CLI Tool. Hier verwenden wir aber nicht das *migrations commands*, sondern das *database update* command.
3. **Per Code**
   Eine dritte Möglichkeit ist die Extension Method auf der DbContext.Database Facade. Diese ist passenderweise Migrate/MigrateAsync betitelt und im Namensraum Microsoft.EntityFrameworkCore zu finden

Zurücksetzen einer Migration

In manchen Fällen ist eine Migration fehlerhaft. Sollte dies der Fall sein, kann man eine Migration zurücksetzen. Dabei gibt es zwei Szenarien:

1. Migration noch nicht auf die Datenbank angewendet
2. Migration bereits auf die Datenbank angewendet

Der 1. Fall ist sehr einfach, man verwendet einfach das command:
dotnet ef migrations remove
und die fehlerhafte Migration wird entfernt.

Der 2. Fall kann schon kniffliger sein, sofern bereits Daten zur Datenbank hinzugefügt wurden.
Grundsätzlich gilt aber, dass wir zuerst die Datenbank zurücksetzen mit
dotnet ef database update <FROM> <TO>
angeschlossen von
dotent ef migrations remove
damit das nächste Mal ausführen der database update Methode nicht wieder zum ausführen der Migration führt.