

Datenbanken Advanced

Übungen Entity Framework – CodeFirst

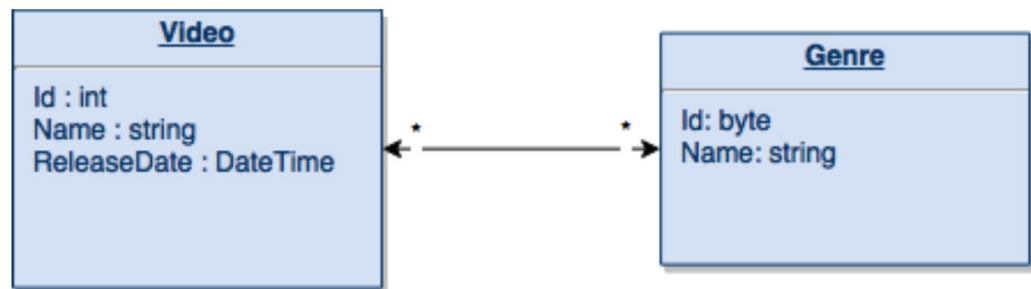
Aufgaben

Ihre Aufgabe ist es, eine Datenanbindung für eine Applikation für Video-Verleih zu erstellen. Die Applikation heisst VidApp. Dabei handelt es sich um eine Konsolenanwendung.

1. Aufgabe

Sie erstellen die Applikation iterativ. In der ersten Iteration soll die Möglichkeit implementiert werden, Videos zur Datenbank hinzuzufügen.

Erstellen Sie eine neue Konsolenanwendung. In dieser erstellen Sie folgendes Model unter Verwendung des «CodeFirst-Workflows»:



Beachten Sie, dass es sich um eine many-to-many-Beziehung handelt zwischen Video und Genre.

Verwenden Sie CodeFirst-Migrationen um die Datenbank zu erstellen und anschliessend eine Datensätze in der Tabelle Genres zu erfassen.

Hinweis: Verwenden Sie zwei Migrationen. Eine für die Erstellung der Tabellen und eine weitere für die Erfassung der Daten in der Tabelle Genres. Grund: Wenn Sie die INSERT INTO Statements in der InitialModel-Migration einfügen und dieses irgendwann z.B. aufgrund von Änderungen überschreiben, verlieren Sie Ihre INSERT INTO Statements.

Prüfen Sie abschliessend die erstellte Datenbank und ihre Struktur.

2. Aufgabe

Sie stellen fest, dass Sie die Lösung «over-engineered» haben. Konkret benötigt jedes Video nur ein Genre.

Ändern Sie Ihr Datenmodel damit jedes Video nur noch ein Genre besitzt.

Verwenden Sie wiederum CodeFirst-Migrationen um die Datenbank zu aktualisieren.

Prüfen Sie abschliessend wiederum Ihre Datenbank, ob die Änderungen korrekt übernommen wurden.

3. Aufgabe

Die Videos sollen zusätzlich in drei Kategorien klassifiziert werden können: Silber, Gold und Platin. Sie entscheiden, dies unter Verwendung eines Enum zu implementieren. Erstellen Sie den Enum Classification und fügen Sie eine Eigenschaft vom Typ Classification zu Ihrer Klasse Video hinzu.

Verwenden Sie wiederum CodeFirst-Migrationen um die Datenbank zu aktualisieren.

Prüfen Sie abschliessend wiederum Ihre Datenbank, ob die Änderungen korrekt übernommen wurden.

4. Aufgabe

Sie sind nun bereit, die Anwendung zu deployen. Ihr DBA erwartet, dass Sie ihm ein Datenbank-Script zur Verfügung stellen. Um ein solches Script mit allen Migrationen zu erstellen, können Sie folgenden Befehl verwenden:

```
PM> Script-Migration -To InitialCreate
```

Dieser Befehl erstellt ein Script, welches alle Migrationen beinhaltet. In einem produktiven Szenario möchten Sie für jedes Deployment genau die entsprechenden Migrationen mitliefern. Dies erreichen Sie mit:

```
PM> Script-Migration -From Migr1 -To Migr2
```

Recherchieren Sie die Befehle sowie die genaue Bedeutung der Parameter.

Lösungen

Siehe 02_EF_CodeFirst_Solution.zip

Denken Sie daran, den ConnectionString entsprechend Ihrer Umgebung anzupassen.

Die Datenbank können Sie folgendermassen aktualisieren:

```
PM> update-database
```