

# Datenbanken Advanced

## Übungen Entity Framework – Queries

### Aufgaben

Verwenden Sie das Projekt in 03\_EF\_Queries.zip um die Queries durchzuführen.

Um die Datenbank zu generieren, führen Sie folgenden Befehl aus:

```
PM> update-database
```

Ggf. müssen Sie EntityFramework deinstallieren und neu installieren. Dies erreichen Sie mit den Befehlen

```
PM> uninstall-package entityframework -force
```

Visual Studio neu starten

```
PM> install-package entityframework -version:6.1.3
```

Evt. tritt beim Update der Datenbank folgender Fehler auf:

```
The object 'DF_Videos_Classifi__2C3393D0' is dependent on column  
'Classification'. ALTER TABLE ALTER COLUMN Classification failed because one  
or more objects access this column.
```

Es handelt sich dabei um einen zufällig generierten Bezeichner (bspw. DF\_Videos\_Classifi\_\_2C3393D0), der auf jeder Datenbank unterschiedlich sein kann. In diesem Fall öffnen Sie die Datenbank VidApp\_Queries in SQL Server Management Studio und erweitern Videos->Constraints. Dort sehen Sie den Constraint. Dieser wurde hinzugefügt, weil Classification keine Nullwerte zulässt. Der Name wurde vom SQL-Server automatisch vergeben. Nun müssen Sie den Namen kopieren und im Migrations-Script EnhanceVideosTable in der folgenden Zeile ersetzen:

```
Sql("ALTER TABLE dbo.Videos DROP CONSTRAINT [constraint-name]");
```

Wenn Sie anschliessend update-database nochmals ausführen, werden die restlichen Scripts fehlerfrei angewendet und die Datenbank steht bereit.

Erstellen Sie nun Queries unter Verwendung von LINQ bzw. LINQ Extension Methods entsprechend den Aufgabenstellungen.

- |                   |   |
|-------------------|---|
| <b>1. Aufgabe</b> | Actionfilme sortiert nach Name.<br><br>Erwartete Ausgabe:<br>Die Hard<br>Terminator 2: Judgment Day<br>The Dark Knight  |
| <b>2. Aufgabe</b> | Gold Drama File sortiert nach Erscheinungsdatum (ReleaseDate). Neuster Film zuerst.<br><br>Erwartete Ausgabe:<br>The Shawshank Redemption<br>Schindler's List |

<b>3. Aufgabe</b>	Alle Filme projiziert in einen anonymen Typ mit den Eigenschaften: MovieName und Genre.
	<p>Erwartete Ausgabe:</p> <p>Anchorman  Die Hard  Schindler's List  The Dark Knight  The Hangover  The Shawshank Redemption  Terminator 2: Judgment Day  The Godfather</p>
<b>4. Aufgabe</b>	Alle Filme gruppiert nach Classification. Die Gruppe soll in einen anonymen Typ mit zwei Eigenschaften projiziert werden: Classification (string) und Movies (IEnumerable<Video>). Für jede Gruppe soll die Classification sowie eine Liste der Videos alphabetisch sortiert angezeigt werden.
	<p>Erwartete Ausgabe:</p> <p>Classification: Silver  Anchorman  Classification: Gold  Die Hard  Schindler's List  The Dark Knight  The Hangover  The Shawshank Redemption  Classification: Platinum  Terminator 2: Judgment Day  The Godfather</p> <p>Hinweis: Verwenden Sie die LINQ Extension Method <code>GroupBy</code>. Nach deren Verwendung müssen Sie für die Projektion <code>Select</code> verwenden. Jede Gruppe hat eine Eigenschaft <code>Key</code> die Sie verwenden können, um die Classification zu erhalten. Die GruppenvARIABLE kann verwendet werden, um alle Videos der Gruppe zu erhalten.</p>
<b>5. Aufgabe</b>	Eine Liste aller Classifications alphabetisch sortiert sowie die Anzahl der zugewiesenen Videos.
	<p>Erwartete Ausgabe:</p> <p>Gold (5)  Platinum (2)  Silver (1)</p>
<b>6. Aufgabe</b>	Liste aller Genres und die Anzahl der zugewiesenen Videos sortiert nach der Anzahl der Videos.
	<p>Erwartete Ausgabe:</p> <p>Action (3)  Drama (3)  Comedy (2)  Horror (0)  Thriller (0)  Family (0)  Romance (0)</p>