



Query Evaluierung

Entity Framework erlaubt uns Queries gegen die Datenbank auszuführen. Mit dem Ergebnis (Resultset) können wir dann in der Applikation Daten anzeigen, verarbeiten oder verändern und in der Datenbank updaten.

Aus Performance Gründen sind hier jedoch folgende Punkte zu beachten:

1. Wir wollen Queries so gut es geht auf der Datenbank durchführen
Queries werden immer dann ausgeführt, wenn wir einen Operator aufrufen, der zur Ausführung bestimmt. Also zum Beispiel:

- ToList, ToArray, ToDictionary
- First, Single, Last
- Count, Any, All

Sollen nun Daten in diesem Resultset **gefiltert, sortiert oder damit gerechnet** werden, sollten wir das tun bevor einer dieser Operatoren aufgerufen wird.

Das hat den entscheidenden Vorteil, dass diese Filter Operationen auf der Datenbank ausgeführt werden. Das ist nicht nur **schneller**, sondern auch **effizienter bezüglich** der zu **übertragenden** Daten.

2. Sollten wir auf Daten nur **lesend** zugreifen sollten wir außerdem **AsNoTracking** verwenden, da dadurch das **ChangeTracking vermieden wird** und wir dementsprechend weniger in Memory overhead haben.
Normalerweise trackt Entity Framework alles was wir von der Datenbank abfragen und hält diesen Tracking Status in Memory, um etwaige Änderungen nachverfolgen und auf die Datenbank schreiben zu können.