

Bulkupsert oder einfach Update, wenn bereits vorhanden ansonsten Insert in MSSQL

Lange habe ich nach einer vernünftigen Lösung gesucht, um riesige Datenmengen schnell in die Datenbank zu schreiben.

Als Quelle steht uns eine `List<Class>` mit der Klasse zur Verfügung, die der Datenbanktabelle gleicht und jede Menge Inhalte enthält. Versucht man diese Liste nun über das Entity Framework in die Datenbank zu jagen, merkt man schnell, dass das Entity Framework an seine Grenzen stößt. Alternativ haben wir das von der `SQLCopy` Klasse die `BulkInsert()` Methode, die aber nur Inserten kann. Findet das `BulkInsert` einen Index/Primärschlüssel, der bereits vorhanden ist, wird die ganze Show abgebrochen. Wie toll wäre es nun, wenn es nicht abbrechen würde, sondern diese Zeile einfach updated. Und das ganze Superschnell. Ich habe Testweise 10.000 Einträge in 400ms und 1.000.000 in 1 min in die Datenbank bekommen.

Methode Upsert:

```
[crayon-676766851c449940623670/]
```

Methode BulkInsert:

```
[crayon-676766851c452633466478/]
```

und die beiden Extension Methods:

```
[crayon-676766851c454040936705/]
```

Angewendet wird das ganze ganz einfach so:

1. Parameter beinhaltet die Liste mit allen Inhalten

2. Parameter gibt an, welche Spalten geprüft werden sollen, ob es zu einem Update kommen soll oder Insert. In meinem Fall, wenn PersNr und CardId bereits in der Datenbank vorhanden sind, so mache ein Update, sonst Insert
3. Parameter gibt an, was inserted werden soll.
4. Parameter gibt an, was geupdated werden soll, wenn Parameter 2 zutrifft.

Bitte auf die tatsächlichen Primärschlüssel, Indizes und NOT NULL Schlüssel/Attribute achten. Ansonsten kommt man schnell zu einem Fehler

[crayon-676766851c456861428916/]