

SQL Server Performanz: Clustered vs. Non-Clustered Index

Kategorie
SQL ServerAaron
Priesterrath

Im diesem Artikel wollen wir uns Indexe und der Auswirkung auf die Performanz des SQL Servers und Abfragen betrachten. Zusätzlich werden wir die beiden unterschiedlichen Index-Typen **Clustered** und **Non-Clustered** vergleichen und ihre unterschiedlichen Einsatzgebiete durchleuchten.

Was ist ein Index?

Die beste Analogie für einen Index ist das Inhaltsverzeichnis eines Buches: Möchte der Leser eine bestimmte Stelle in einem Buch finden, kann er das gesamte Buch durchblättern bis er die richtige Stelle gefunden hat. Alternativ könnte er das Inhaltsverzeichnis verwenden um direkt in den richtigen "Bereich" des Buches zu gelangen, an dem sich die Stelle befindet, für die er sich interessiert. Ähnlich funktioniert ein Index für eine Tabelle: es wird eine Art "Inhaltsverzeichnis" auf einer bestimmten (oder mehreren) Spalte der

Tabelle erzeugt um Abfragen bezüglich dieser Spalte schneller verarbeiten zu können.

Beispiel

Um die Vorteile eines Index weiter zu verdeutlichen, betrachten wir folgendes, simples Beispiel:

Es existiert eine Tabelle "Cars", die Informationen bezüglich Autoverkäufen bereitstellt. Eine Spalte der Tabelle, "car_color", beschreibt die Farbe des verkauften Autos. Auf ihr existiert ein beliebiger Index. An dieser Stelle kann der Index als eine Art der Gruppierung betrachtet werden.

Nun betrachten wir eine Abfrage, welche alle verkauften Autos der Farbe Blau ausgibt. Ohne einen Index müsste die gesamte Tabelle durchsucht und die Farbe von jedem verkauften Auto betrachtet werden.

Mit Hilfe des Indexes sind die Einträge der Tabelle sozusagen sortiert. Der "Block", in dem sich die blauen Autos befinden, muss lediglich identifiziert und kann anschließend direkt zurück gegeben werden.

Index-Typen

Im Allgemeinen wird meist zwischen den beiden Index-Typen **Clustered** und **Non-Clustered** (manchmal auch **Unclustered** genannt) unterschieden.

Clustered Index

Ein **Clustered Index** ist ein Index, in dem die tatsächliche Anordnung der Tabellen-Zeilen im Speicher verändert wird. Wird ein Clustered Index auf einer Spalte erzeugt, wird die zugehörige Tabelle, basierend auf der ausgewählten Spalte, physisch sortiert. Diese Sortierung ist vergleichbar mit einem Wörterbuch.

```
CREATE CLUSTERED INDEX [INDEX_NAME] ON [TABLE_NAME] ([COLUMN_NAME] ASC)
```

Mit Hilfe dieses t-SQL Befehls kann ein Clustered Index auf einer Tabelle für eine bestimmte Spalte erzeugt werden. Der **INDEX_NAME**, **TABLE_NAME** und **COLUMN_NAME** muss jedoch individuell angepasst werden. Eine beispielhafte Eingabe für das oben angegebene Beispiel der Autoverkäufe könnte wie folgt aussehen:

```
CREATE CLUSTERED INDEX [IX_CLUSTERED_CAR_COLOR] ON [dbo].[Cars] ([car_color] ASC)
```

Non-Clustered Index

Ein **Non-Clustered Index** verändert die Anordnung der Tabellen-Zeilen im Speicher nicht. Statt dessen wird ein neues Objekt erzeugt, welches die für den Index ausgewählte Spalte beinhaltet und sortiert ist. Zusätzlich gibt es Zeiger, die auf die zugehörige Zeile in der tatsächlichen Tabelle verweisen.

Der **Non-Clustered Index** ist vergleichbar mit einem Index auf der letzten Seite eines Buches: Alle im Buch auftretenden Begriffe werden in alphabetischer Reihenfolge dargestellt und die Stelle im Buch angegeben, an der der Begriff im Buch zu finden ist.

Nachdem das zusätzliche Objekt erzeugt wurde, wird dieses mit Hilfe eines B-Baums effizient repräsentiert.

```
CREATE NONCLUSTERED INDEX [INDEX_NAME] ON [TABLE_NAME] ([COLUMN_NAME] ASC)
```

Mit Hilfe dieses t-SQL Befehls kann ein Non-Clustered Index auf einer Tabelle für eine bestimmte Spalte erzeugt werden. Der **INDEX_NAME**, **TABLE_NAME** und **COLUMN_NAME** muss jedoch individuell angepasst werden. Eine beispielhafte Eingabe für das oben angegebene Beispiel der Autoverkäufe könnte wie folgt aussehen:

```
CREATE NONCLUSTERED INDEX [IX_NONCLUSTERED_CAR_COLOR] ON [dbo].[Cars] ([car_color] ASC)
```

Relation der Index-Typen

Eine Besonderheit der Index-Typen ist das Zusammenspiel der Indexe, wenn ein Clustered und ein Non-Clustered Index verwendet wird.

Für gewöhnlich wird der Zeiger, der auf die zugehörige Zeile in der Tabelle zeigt, über die **rowID (RID)** realisiert. Existiert auf einer Tabelle jedoch bereits ein Clustered Index, wird stattdessen auf den Index-Schlüssel des Clustered Index zurückgegriffen.

Vor- und Nachteile eines Indexes

Im Allgemeinen beschreibt ein Index den Tausch von Speicher gegen erhöhte Performanz: Ein zusätzliches Objekt muss angelegt oder die Tabelle erweitert werden, dafür können Abfragen schneller verarbeitet werden. Eine wichtige Veränderung des Systems darf dabei jedoch nicht übersehen werden. Verändert sich der Datenbestand der Tabelle (durch Insert, Update, Merge oder Delete), müssen die Änderungen auch auf den Index übertragen werden.

Im Beispiel einer Insert-Operation auf einer Tabelle, die einen Clustered Index besitzt, muss also nach dem Einfügen der neuen Informationen der Index neu sortiert werden. Dies kann eine kostspielige Operation sein!

Im Vergleich dazu kann die Nutzung von beiden Index-Typen einen positiven Effekt auf Insert-Operationen haben. Mit Hilfe der Darstellung der Reihenfolge durch den Non-Clustered Index und die Nutzung des sortierten Schlüssels des Clustered-Index werden aufwendige Reorganisationsprozesse umgangen. Dies bringt jedoch eine erhöhte Latenz in Abfragen, die mehr Spalten, als nur die im Index enthaltenen, anspricht, mit sich.

Fazit

Ein Index ist ein mächtiges Werkzeug, um die Performanz bezüglich bestimmter Abfragen zu verbessern. Dafür müssen jedoch die richtigen Mittel (genügend Speicher) verfügbar sein. Den richtigen Index für eine Tabelle zu finden ist nicht immer trivial und sollte mit hoher Sorgfalt und regelmäßiger Überprüfung durchgeführt werden.