

FOM MÜNCHEN

FACHBEREICH WIRTSCHAFTSINFORMATIK

**Seminararbeit**

# **Optimierung von MySQL Anfragen unter Zuhilfenahme von Explain**

Eingereicht von:

Oliver Kurmis

Perfallstraße 8 81675 München Email: [oliver@kurmis.com](mailto:oliver@kurmis.com)

Abgegeben am:

1. Juli 2014

Erarbeitet im:

3. Semester

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>III</b>
<b>Tabellenverzeichnis</b>	<b>III</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Problemstellung . . . . .	1
1.2 Zielsetzung der vorliegenden wissenschaftlichen Auseinandersetzung . .	1
1.3 Vorgehensbeschreibung . . . . .	1
<b>2 Theoretische Grundlagen</b>	<b>2</b>
2.1 Der physische Zugriff auf die Daten . . . . .	2
2.2 Speicherstrukturen . . . . .	2
2.3 Bearbeitung von SQL-Statements . . . . .	2
2.4 Optimierungen : frühzeitige Restriktionen, JOINS . . . . .	2
<b>3 Beispiel MySQL</b>	<b>2</b>
3.1 einfache Select-Anfragen (eine Tabelle) . . . . .	3
3.2 Umschreiben von Nicht-Select-Anfragen . . . . .	3
3.3 Die Spalten der EXPLAIN-Ausgabe . . . . .	3
3.3.1 EXPLAIN EXTENDED . . . . .	6
3.3.2 EXPLAIN PARTITIONS . . . . .	6
3.4 Abfragen mit mehreren Tabellen . . . . .	6
3.5 Optimierungsmöglichkeiten und Benchmarking . . . . .	6
3.6 Visuelles EXPLAIN (graphische Werkzeuge) . . . . .	6
<b>4 Fazit und Ausblick</b>	<b>6</b>
<b>5 Anhang</b>	<b>7</b>
<b>Literatur</b>	<b>8</b>

## Abkürzungsverzeichnis

**CPU**      Central Processing Unit (deutsch: Hauptprozessor)

**DB**        Datenbank

**RAM**      Random Access Memory (deutsch: Hauptspeicher oder Arbeitsspeicher)

**RDBMS**   Relationales Datenbank-Managementsystem

**SQL**       Structured Query Language

## Abbildungsverzeichnis

## Tabellenverzeichnis

# 1 Einleitung

## 1.1 Problemstellung

Datenbank-Systeme finden heute in nahezu allen IT-Systemen Verwendung. Der Optimierung von Datenbank-Anfragen kommt daher eine große Bedeutung zu. Hierfür gibt es eine Vielzahl von Möglichkeiten, z.B. Latenz und Bandbreite der Anbindung der Datenbank, Leistungsfähigkeit des Datenbank-Servers, Anzahl der Datenbank-Anfragen im Programmcode, Cachingmechanismen.

Hat man andere Flaschenhälse ausgeschlossen oder bereits optimiert, gilt es die relevanten SQL-Abfragen des Systems zu identifizieren und gezielt zu optimieren.

Viele RDBMS stellen mit dem SQL-Kommando EXPLAIN eine Möglichkeit zur Verfügung, mehr über die innere Arbeitsweise der Datenbank bei einer bestimmten SQL-Abfrage zu erfahren. Durch gezielte Veränderung der SQL-Abfrage oder des Datenschemas kann somit die Bearbeitung der Abfrage optimiert werden.

## 1.2 Zielsetzung der vorliegenden wissenschaftlichen Auseinandersetzung

Die folgende Arbeit bezieht sich speziell auf die Optimierung von SQL-Anfragen mittels EXPLAIN bei dem RDBMS MySQL. Es soll untersucht werden

## 1.3 Vorgehensbeschreibung

-Literaturrecherche betreiben, -wesentliche Punkte zusammengefasst -an Beispieldatenbank experimentell nachvollzogen

## 2 Theoretische Grundlagen

### 2.1 Der physische Zugriff auf die Daten

Daten einer DB werden in der Regel auf einer Festplatte (HDD) oder einem Flash-Laufwerk (SSD) gespeichert. Das RDBMS nutzt dazu Funktionen des Betriebssystems auf verschiedenen Ebenen. Dateisystem-Treiber, nimmt Lese und Schreib Anforderungen für Datensätze an und gibt rechnet diese in nie durchnummerierten Blöcke des Blockgerätes um. Der Blockgeräte-Treiber liest dann die entsprechenden Blöcke von der Platte oder schreibt sie.

DBMS -> Dateisystemtreiber -> Blockgerätetreiber -> HDD/SSD

HDD-Blockgerätetreiber-Dateisystemtreiber-DBMS

### 2.2 Speicherstrukturen

Binärbaum:

B-Baum, Hashing, Heap

### 2.3 Bearbeitung von SQL-Statements

Umsetzung in relationale Algebra

### 2.4 Optimierungen : frühzeitige Restriktionen, JOINS

## 3 Beispiel MySQL

In vielen Datenbanken steht mit dem SQL-Kommando EXPLAIN ein Werkzeug zur Verfügung, um mehr darüber zu erfahren, wie eine Datenbank eine bestimmte Anfra-

ge ausführt, wie also der Every Execution Plan (QEP) ist. Das EXPLAIN-Kommando gehört jedoch nicht zum SQL-Standard und wird bei den verschiedenen RDBMS unterschiedliche Ausgaben erzeugen. Bei MySQL ist EXPLAIN sehr mächtig und gibt umfassend und datailliert Auskunft über den QEP.

### 3.1 einfache Select-Anfragen (eine Tabelle)

### 3.2 Umschreiben von Nicht-Select-Anfragen

As of MySQL 5.6.3, permitted explainable statements for EXPLAIN are SELECT, DELETE, INSERT, REPLACE, and UPDATE. Before MySQL 5.6.3, SELECT is the only explainable statement. [4]

### 3.3 Die Spalten der EXPLAIN-Ausgabe

Als Ergebnis einer EXPLAIN-Anfrage liefert MySQL eine Tabelle mit festen Spalten und einer oder mehrerer Zeilen, je nach Komplexität der Anfrage. Die einzelnen Spalten haben folgende Bedeutung:

#### **id**

Diese Zahl identifiziert das SELECT, zu dem die Zeile gehört. Bei einer einfachen SELECT-Abfrage steht in diesem Feld immer nur die 1.

#### **select\_type**

Die Spalte gibt an, ob es sich um ein einfaches oder komplexes SELECT handelt. Folgende Werte können hierbei auftreten:

**SIMPLE** einfaches SELECT, keine Unterabfragen oder UNIONS

**PRIMARY** äußeres SELECT eines komplexen SELECT

**SUBQUERY** SELECT in einer Unterabfrage

**DERIVED** SELECT in einer Unterabfrage in der FROM-Klausel

**UNION** zweites bzw. nachfolgende SELECT einer UNION

**UNION RESULT** Ergebnis des UNION, wird aus temporärer Tabelle geholt

## **table**

auf welche Tabelle wird zugegriffen

Tabellenname oder Alias

Spalte von oben nach unten lesen

## **type**

Wie ist der Zugriffstyp, wie wird MySQL die Zeilen in der Tabelle auffinden? Folgende Werte können hierbei auftreten (in geordneter Reihenfolge vom langsamsten zum schnellsten Zugriffstyp):

**ALL** Tablescan, d.h. Tabelle muss in der Regel von Anfang bis Ende durchlaufen werden, ist ein starker Indiz für weiteren Optimierungsbedarf.

**index** wie Tablescan, aber in Indexreihenfolge, eine Sortierung wird hierbei vermieden.

**range** Bereichsscan, d.h. eingeschränkter Indexscan, z.B. bei BETWEEN oder WHERE  
x >

**ref** Indexzugriff findet statt, auch Index-Lookup genannt, Zeilen entsprechen einem Wert, nur bei einem nichteindeutigen Index, Index wird hierbei mit einem Referenzwert verglichen. Variante: ref\_or\_null

**eq\_ref** Index-Lookup mit eindeutigem Treffer, bei Primärschlüssel oder eindeutigem Index

**const,system** der Datenzugriff konnte von MySQL wegoptimiert oder in eine Konstante umgewandelt werden.

**NULL** Abfrage kann von MySQL bei der Optimierung aufgelöst werden, kein Zugriff auf Tabelle oder Index, z.B. Minimum einer indizierten Spalte

## **possible\_keys**

Diese Spalte gibt an, welche Indizes für die Bearbeitung der Anfrage prinzipiell zur Verfügung stehen. Die hier stehenden Werte werden bereits in einer frühen Phase der Optimierung ermittelt, letztendlich wird in der Regel nur ein Index genutzt. Sind hier viele Indizes aufgeführt deutet das auf ein Problem hin.

## **key**

Die Spalte key gibt an, welcher Index der Optimierer für die Anfrage gewählt hat. Dies kann auch ein abdeckender Index sein, aus dem die Ergebniswerte gelesen werden können ohne dass die eigentliche Tabelle gelesen werden muss. Ein Wert von NULL in der Spalte key bedeutet, dass kein Index genutzt wird und ist ein starkes Indiz für einen Optimierungsbedarf.

**key\_len**

wie viel Byte (Spaltenbreite) eines Index werden benutzt welche Spalten des Index werden genutzt, von links beginnend

**ref**

-welche Spalten aus früheren Tabellen werden benutzt, um in dem key-Index nachzuschlagen

**rows**

Die Zahl in der Spalte rows ist eine Schätzung für die Anzahl der Zeilen, die gelesen werden müssen. Bei Abfragen mit mehreren Tabellen bezieht sich diese Angabe pro Schleife im Nested-Loop-Join-Plan. Die Schätzung beruht auf Statistiken kann ungenau sein. Im besten Fall steht hier eine 1.

**filtered**

Die Spalte ist neu seit MySQL 5.1 und erscheint nur bei EXPLAIN EXTENDED. Es ist eine pessimistische Schätzung des Prozentsatzes der Zeilen,...

**Extra**

**Using Index** abdeckender Index wird genutzt, d.g. die angefragten Daten müssen nicht aus der Tabelle gelesen werden

**Using where** die Zeilen werden nachträglich gefiltert, d.h. für die WHERE-Bedingung wird nicht der Index genutzt

**Using temporary** Erstellung einer temporäre Tabelle für Sortierung

**Using filesort** externe Sortierung, im RAM oder auf den Datenträger

**ranke checked for each record (index map: N)** kein geeigneter Index vorhanden

vgl. [5]

### 3.3.1 EXPLAIN EXTENDED

Extra Spalte filtered.



### **3.3.2 EXPLAIN PARTITIONS**

Seit Mysql 5.1, extra Spalte Partitions

## **3.4 Abfragen mit mehreren Tabellen**

## **3.5 Optimierungsmöglichkeiten und Benchmarking**

Wichtigsten Spalten key (benutzer Index), rows (Anzahl Zeilen bearbeitet), type

QEP ist nicht fix, wird bei jeder Abfrage neu erstellt. Daher EXPLAIN einer Abfrage mit verschiedenen Beispielwerten. Nicht mit Test-Daten testen, sondern RealWord-Daten, z.B. Backups.

```
ALTER TABLE users ADD INDEX
```

Vorsicht beim Anlegen von INDIZES: Kann bei grossen Tabellen sehr lange Dauern und blockiert in dieser Zeit die Tabelle. Vorher Informationen über die Grösse der Tabellen holen, am besten einen Probedurchlauf auf einer Kopie machen.

## **3.6 Visuelles EXPLAIN (graphische Werkzeuge)**

## **4 Fazit und Ausblick**

Beschränkungen! Optimierung wichtig Mit Explain möglich nicht immer exakte Angaben Kontrolle der Optimierung mit Benchmarks nötig möglichst bereits in den Entwicklungsprozess integrieren, und nicht erst wenn es brennt

## 5 Anhang

Was so alles in einen Anhang kommt

# Literatur

- [1] Schwartz, B., Zaitsev, P., Tkachenko, V., Zawodny, J.D., Lentz, A., Balling, D.J. (2009) *High Performance MySQL. Optimierung, Datensicherung, Replikation & Lastverteilung*, 2. Auflage, O'Reilly Verlag, 2009
- [2] Sauer, H. (1998) *Relationale Datenbanken, Theorie und Praxis*, 4. Auflage, Addison Wesley Longman Verlag, 1998
- [3] Bradford, R. (2011) *Effective MySQL: Optimizing SQL Statements*, McGraw-Hill Osborne Media, 2011
- [4] *Optimizing Queries with EXPLAIN* <http://dev.mysql.com/doc/refman/5.6/en/using-explain.html>
- [5] *EXPLAIN Output Format* <http://dev.mysql.com/doc/refman/5.6/en/explain-output.html>

# Ehrenwörtliche Erklärung

Hiermit versichere ich, dass die vorliegende Arbeit von mir selbstständig und ohne unerlaubte Hilfe angefertigt worden ist, insbesondere dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen sind, durch Zitate als solche gekennzeichnet habe. Ich versichere auch, dass die von mir eingereichte schriftliche Version mit der digitalen Version übereinstimmt. Weiterhin erkläre ich, dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat. Ich erkläre mich damit einverstanden/nicht einverstanden, dass die Arbeit der Öffentlichkeit zugänglich gemacht wird. Ich erkläre mich damit einverstanden, dass die Digitalversion dieser Arbeit zwecks Plagiatsprüfung auf die Server externer Anbieter hoch geladen werden darf. Die Plagiatsprüfung stellt keine Zurverfügungstellung für die Öffentlichkeit dar.

München, 30. Juni 2014



Oliver Kurmis