

# MySQL New Features 5.6

**Oli Sennhauser**  
**FromDual GmbH**  
**Uster / Schweiz**

## Schlüsselworte

MySQL, 5.6, New Features, InnoDB, Partitionen, Memcached, NoSQL, Performance

## Einleitung

Vor circa zwei Jahren hat Oracle den MySQL Release 5.5 für die Produktion freigegeben. Dieser Release stammte hauptsächlich noch aus der Feder von MySQL/Sun.

Worauf Oracle bei der Weiterentwicklung von MySQL Wert gelegt hat, und welche spannenden Verbesserungen in den neuen Release mit eingeflossen sind werden wir im Folgenden beleuchten.

## Verbesserungen durch den neuen Release

InnoDB wird von Oracle bevorzugt behandelt. Dies hat sich bereits darin gezeigt, dass InnoDB mit MySQL 5.5 zur default Storage Engine erklärt wurde. Mit den zahlreichen Verbesserungen in diesem Bereich macht Oracle deutlich, dass der eingeschlagene Weg beibehalten wird.

Dies manifestiert sich darin, dass Oracle zum Beispiel die Volltext-Suche auch für InnoDB implementiert, weiter massiv an der Performance-Schraube gedreht hat und auch in Richtung Data Warehousing einige interessante Funktionalitäten hinzugekommen sind.

Auch bei der Partitionierung, der Replikation und der Sicherheit sind zahlreiche Neuerungen mit eingeflossen. Und nicht zu vernachlässigen sind die weit über 400 Bugs und kleinen Unzulänglichkeiten welche im neuen Release behoben und verbessert wurden. Leider sind die Änderungen so zahlreich, dass wir uns hier auf ein paar wenige spannende beschränken müssen...

## InnoDB Volltext-Suche

Die Volltext-Suche wird dann benötigt, wenn ein Wort innerhalb eines Textes gesucht werden soll.

Wer bereits einmal versucht hat, auf einer sehr grossen Tabelle eine Abfrage des folgenden Typs abzusetzen, wird eine Weile auf das Resultat gewartet haben:

```
SELECT titel FROM article WHERE article LIKE '%MySQL%';
```

Der Grund, warum dies Art von Abfragen lange dauert, liegt darin, dass das Datenbanksystem den ganzen Index auf `article` durchlesen muss um nachzuschauen, ob da "MySQL" drin steht. Liegen die Daten auf Platte anstatt im Speicher vor, kann das schon ein Weilchen dauern.

Genau hier kommt die Volltext-Suche ins Spiel. Bis anhin war die Volltext-Suche für InnoDB-Tabellen nicht verfügbar. Dies hatte zur Folge, dass man entweder auf die nicht Crash-sicheren MyISAM Tabellen ausweichen oder auf externe Lösungen zur Volltext-Indexierung zurückgreifen musste.

Neu kann man bei InnoDB jetzt einen Volltext-Index erstellen und dann die Daten entsprechend effizienter abfragen:

```
ALTER TABLE article ADD FULLTEXT INDEX (title);
```

```
SELECT titel FROM article WHERE MATCH (data) AGAINST ('MySQL');
```

Die Abfrage sieht leicht anders aus, als obige, das Resultat ist aber das selbe. Und die Ausführungszeit sollte signifikant geringer ausfallen.

## Online Schema-Änderungen

Die zunehmend gestiegenen Anforderungen an die Verfügbarkeit haben auch bei MySQL nicht halt gemacht. Bis anhin musste man sich mit verschiedenen Tricks behelfen, wenn man die Datenbank auch bei Schema-Änderungen online halten wollte.

Neu macht MySQL uns diese Sache wesentlich einfacher. Mit MySQL 5.6 sind zahlreiche Funktionen jetzt online, das heisst, während des laufenden Betriebs (lesend UND schreibend), verfügbar. Folgende Funktionen gehören unter anderen dazu: ADD INDEX, DROP INDEX, das Ändern des AUTO\_INCREMENT Wertes, das Hinzufügen und Löschen von FOREIGN KEY CONSTRAINTS oder das Umbenennen einer Spalte.

Früher haben alle dies Operationen ein Umkopieren der Daten erfordert. Heute können Operationen wie das Löschen, Hinzufügen, oder das Ändern der Reihenfolge von Spalten zwar auch online durchgeführt werden, erfordern aber immer noch ein Umkopieren der Daten im Hintergrund.

## Transportierbare Tablespaces

Ein Grund, warum sich einige MySQL Nutzer bis anhin gegen einen Wechsel zu InnoDB gestäubt haben ist, dass sich InnoDB Tabellen nicht auf Datei-Ebene von einer MySQL Instanz in eine andere kopieren lassen.

Dies ist insbesondere bei Data Warehouses eine gängige Praxis um effizient Daten vom operativen System in die Staging Area und weiter in den Data Marts zu transportieren.

Mit MySQL 5.6 ist nun auch diese Restriktion gefallen Mit den Befehlen FLUSH TABLE FOR EXPORT und IMPORT TABLESPACE können InnoDB Tabellen einfach und sicher zwischen MySQL Instanzen ausgetauscht werden, was bis anhin nur mit MyISAM Tabellen möglich war.

Als erstes wir das Ziel vorbereitet:

```
CREATE TABLE test (...) ENGINE = InnoDB;  
ALTER TABLE test DISCARD TABLESPACE;
```

Dieser Schritt mutet etwas holprig an und es ist zu hoffen, dass er in einer späteren Version nicht mehr benötigt wird.

Anschliessend wird die Tabelle von der Quelle um Ziel transportiert:

```
FLUSH TABLE test FOR EXPORT;  
\! scp $datadir/test/test.ibd mysql@remote:/var/lib/mysql/test/  
\! scp $datadir/test/test.cfg mysql@remote:/var/lib/mysql/test/  
UNLOCK TABLES;
```

Während dieser Operation sind zwar noch Lesezugriffe auf die Tabelle zulässig, aber nicht mehr Schreibzugriffe. Es ist anzunehmen, dass diese Einschränkung früher oder später ebenfalls den online DDL Verbesserungen zum Opfer fallen wird.

Als letzter Schritt muss der so transportierte InnoDB Tablespace noch in die Ziel-Datenbank importiert werden:

```
ALTER TABLE test IMPORT TABLESPACE;
```

Und schon stehen die Daten für den Zugriff zur Verfügung.

Das Ganze funktioniert aber nur, wenn jede Tabelle in einem eigenen Tablespace vorliegt, also `innodb_file_per_table` beim Anlegen der Tabelle auf 1 gestellt war.

Mit Mysql 5.6.6 hat Oracle aber die default Einstellungen für `innodb_file_per_table` sowieso auf 1 geändert. Somit stellt das nicht weitere eine grosse Hürde dar.

Im Bereich Tablespace Mangement ist weiter hinzugekommen ,dass man neu Tabellen beim Erstellen mit dem Zusatz `DATA DIRECTORY='/media/disk/ssd/'` spezifisch auf bestimmte Platten legen kann. Was zum Beispiel bei Tabellen sinnvoll ist, auf denen stark random gelesen oder geschrieben wird. Diese Tabellen können dann zum Beispiel auf SSD-Platten gelegt werden.

Auch der das UNDO Log kann neu in einen eigen Tablespace und allenfalls ebenfalls auf eine SSD-Platte gelegt werden.

### Partitionen

Auch beim Partitionieren von Tabellen hat sich etwas getan. Partitionen werden meist dann verwendet, wenn sehr grosse Datenmengen im Spiel sind. Zum Beispiel bei Data Warehouses. Sie dienen einerseits der besseren Handhabung der Daten, andererseits bringen Partitionen, richtig eingesetzt, auch Performance Vorteile mit sich.

Eine Funktionalität die besonders ins Auge sticht, ist das Austauschen einer Tabelle mit der entsprechenden Partition. Dies wird mit dem Befehl: `ALTER TABLE .. EXCHANGE PARTITION` erreicht.

Spannend wird das Ganze, wenn wir effizient Daten verschieben wollen. Unter Zuhilfenahme der weiter oben beschriebenen transportierbaren Tablespaces lassen sich so recht einfach Daten von der Staging Area in den Data Mart verschieben:

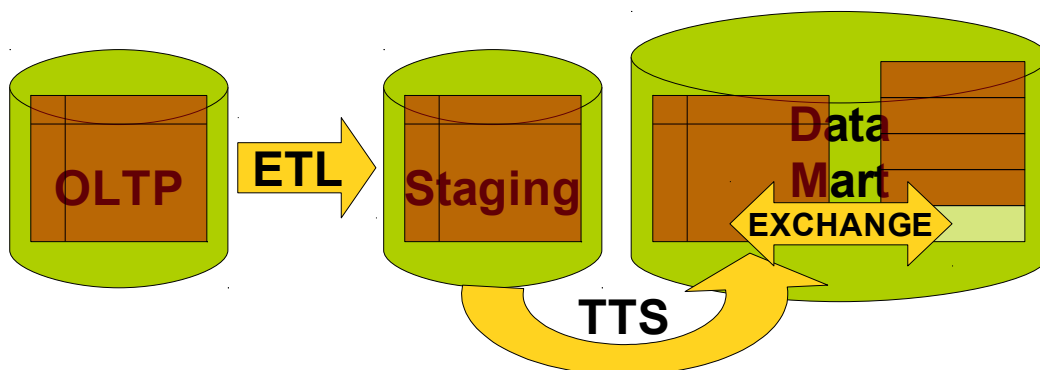


Abb. 1: Transportieren und Austauschen von InnoDB Tablespaces

### NoSQL Technologie mit Memcached

Mit der Version 5.6 bietet MySQL neu auch eine NoSQL Schnittstelle auf die InnoDB Storage Engine an. Bei sehr kurzen Abfragen im Bereich von wenigen Millisekunden nimmt der Overhead, welcher der SQL-Teil einer Abfrage ausmacht (Zugriffsrechte, Parsen und Optimieren der Abfrage, etc.) bis zum 80% der Ausführungszeit in Anspruch. Wenn dieser Teil weg optimiert werden kann, wird sowohl die Antwortzeit signifikant verkürzt als auch der Durchsatz erhöht. Genau hier setzt diese Technologie an: Anstatt über SQL Befehle auf die Daten zuzugreifen wird über das Memcached Plugin direkt in die Storage Engine hinein gegriffen um an die Daten zu gelangen:

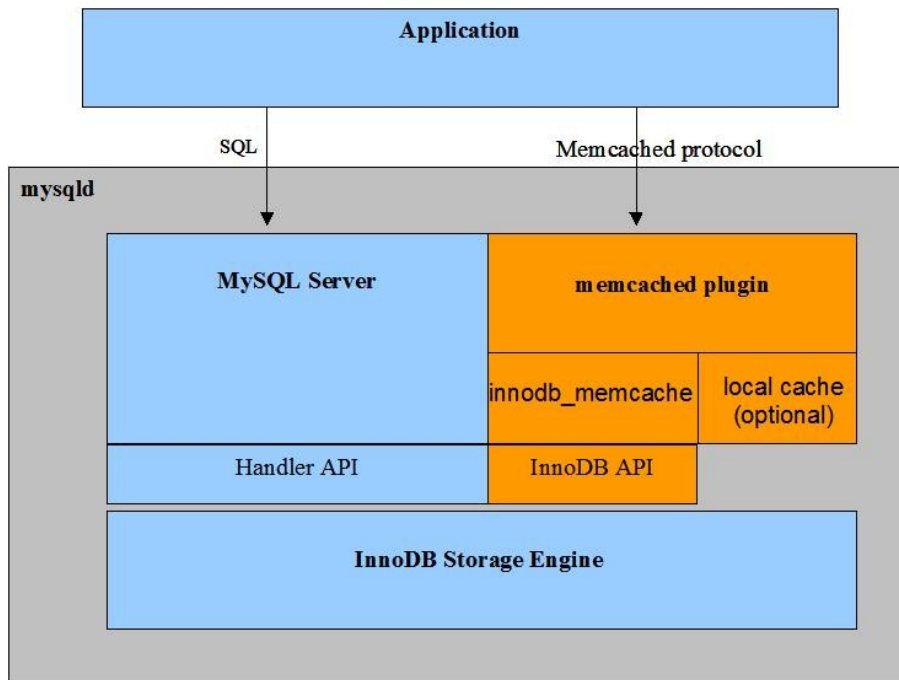


Abb. 2: Memcached Plugin als NoSQL Technologie

Die Befehle um Daten zu ändern und abzufragen sehen dementsprechend auch etwas anders aus:

```
shell> telnet 127.0.0.1 11211
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.

telnet> add abcd 0 900 6
blabla
STORED

telnet> get abcd
VALUE abcd 0 6
blabla
END
```

Nichts desto trotz kann man auf dies Daten auch konventionell über das SQL-Interface zugreifen um zum Beispiel Reports zu generieren:

```
SELECT * FROM test.demo_test;
+-----+-----+-----+-----+-----+
| c1   | c2           | c3   | c4   | c5           |
+-----+-----+-----+-----+-----+
| abcd | blabla      | 0    | 1    | 1346833525  |
+-----+-----+-----+-----+-----+
```

Diese und viele weitere Verbesserungen sind in MySQL 5.6 eingeflossen. Leider können wir hier nur einige wenige dieser Änderungen genauer betrachten...

**Kontaktadresse:**

Oli Sennhauser  
FromDual GmbH  
Rebenweg 6  
CH – 8610 Uster

Telefon: +41 44 – 940 24 82  
Fax: +41 43 – 55 68204  
E-Mail: [oli.sennhauser@fromdual.com](mailto:oli.sennhauser@fromdual.com)  
Internet: [www.fromdual.com](http://www.fromdual.com)