

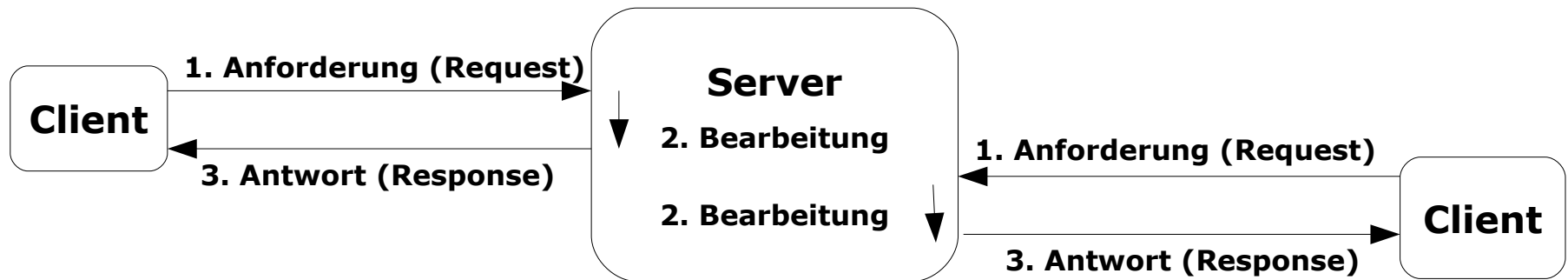
## **MySQL-Clients**

### **Colletta Hagert**

## Einführung

## Client-Server Architektur

Die meisten heutigen Datenbanksysteme arbeiten nach dem Client-Server-Konzept.



## Client-Server Architektur

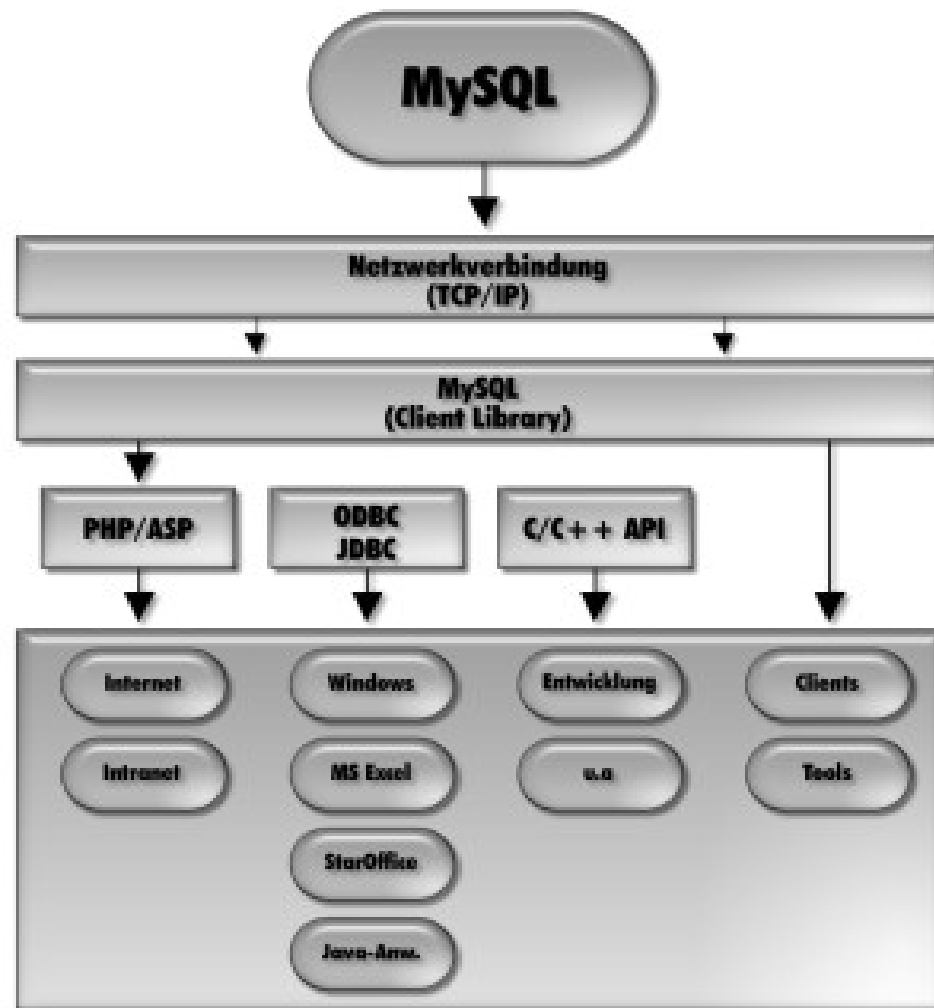
MySQL ist ein Client-Server-System.

=> Auf den MySQL-Server können dabei beliebig viele Clients zugreifen bzw. Anfragen an ihn richten.

Clientprogramme (**MySQL-Client**, PHP, JSP (Java Server Pages), JDBC (Java DataBase Connectivity), Access)) müssen sich nur auf irgendeine Art und Weise über die Schnittstellen, die MySQL bietet, verständigen können.

Die Clients können dabei auf demselben Rechner installiert sein wie der Server. Alternativ können sie über eine TCP/IP-Netzwerkverbindung Verbindung aufnehmen, soweit die entsprechenden Zugriffsrechte vorhanden sind.

## Überblick über das Schnittstellenkonzept von MySQL



## MySQL Clients

MySQL-Clients stellen die Verbindungen mit dem Server her, um auf Datenbanken zuzugreifen oder administrative Aufgaben durchzuführen.

Umgebungsvariablen, die von MySQL-Clientprogramme verwendet werden sind:

**MYSQL\_UNIX\_PORT:** die vorgabeseitige Unix-Socketdatei. Sie wird für Verbindungen mit localhost verwendet.

**MYSQL\_TCP\_PORT:** die Standardportnummer. Wird für TCP/IP-Verbindungen verwendet.

**MYSQL\_PWD:** das Standardpasswort.

**MYSQL\_DEBUG:** gibt Trace-Optionen für das Debuggen an.

**TMPDIR:** Verzeichnis, in dem Temporärtabellen und -dateien erstellt werden.

- **mysql:**

Befehlszeilen-Tool zur interaktiven Eingabe von SQL-Anweisungen und zur Ausführung solcher Anweisungen aus einer Datei im Stapelbetriebsmodus

- **mysqladmin:**

Führt administrative Operationen durch.

- **mysqlcheck:**

Wird zur Wartung von Tabellen verwendet. Überprüft, repariert, analysiert und optimiert Tabellen.



- **mysqldump:**

Speichert eine MySQL-Datenbank in Form von SQL-Anweisungen oder als tabulatorgetrennte Texte in eine Datei.

- **mysqlimport:**

Importiert Textdateien mithilfe von LOAD DATA INFILE in die jeweiligen Tabellen.

- **mysqlshow:**

Zeigt Informationen zu Datenbanken, Tabellen, Spalten und Indizes an.

**mysql**

- Eine einfache SQL-Shell.
- Unterstützt **Optionen**, die auf Kommandozeile spezifiziert werden können
- Kann bestimmte **Befehle** selbst interpretieren
- Zwei Möglichkeiten mysql einzusetzen sind:

## **1. Interaktive:**

Die Abfrageergebnisse werden im ASCII-Tabellenformat angezeigt.

## **2. Nicht interaktive:**

Das Ergebnis durch Tabulatoren getrennt aufgelistet.

Das Ausgabeformat kann mit bestimmten Optionen (z. B, -X) beeinflusst werden. -X erzeugt eine XML-formatierte Ausgabe.

## Aufruf von mysql

Die mysql-Shell verfügt über eine Reihe von Parametern für den Aufruf. Diese können wie folgt angezeigt werden:

```
shell> mysql --help
```

## Beispiele

```
shell> mysql Datenbankname
```

Oder:

```
shell> mysql --user=username --password=passwort Datenbankname
```

- Eine SQL-Anweisung muss auf `;`, \g oder \G enden.

## SQL-Befehle aus einer Textdatei laden:

Es ist allerdings auch möglich, SQL-Anweisungen in einer Datei abzulegen und mysql dann anzuweisen, seine Eingabe aus dieser Datei auszulesen:

1. Ein Textdatei erstellen, die die Anweisungen enthält, die ausgeführt werden sollen: Textdatei
2. mysql wie folgt aufrufen:

```
shell> mysql Datenbankname < Textdatei
```

Oder

```
shell> mysql < Textdatei
```

falls als erstes eine *USE Datenbankname*-Anweisung in die Datei gesetzt ist.

## SQL-Befehle aus einer Textdatei laden:

Wird mysql bereits ausgeführt, dann können Sie eine SQL-Skriptdatei mithilfe des Befehls *source* oder `\.` ausführen:

```
mysql> source Dateiname.sql
```

oder

```
mysql> \. Dateiname.sql
```

**mysqladmin**

Wird verwendet, um die administrative Aufgaben durchzuführen, z. B.,

- Überprüfen der Konfiguration und des aktuellen Status des Servers
- Erstellen und löschen von Datenbanken
- den Server herunterfahren,
- Datenbanken anlegen,
- Datenbanken löschen,
- Versions-, Prozess- und Statusinformationen anzeigen,
- Rechtetabellen neu laden.



## Verwendung von mysqladmin

```
shell> mysqladmin [options] command [command-arg] [command  
[command-arg]] ...
```

=> Einige Befehle nehmen auf den Befehlsnamen folgend ein Argument entgegen.

=> [...] steht für optional

## Beispiele

```
shell> mysqladmin --help
```

```
shell> mysqladmin -uroot -hhostname password "Neues Passwort"
```

=> Der Befehl *password* erwartet als Argument ein neues Passwort

```
shell> mysqladmin -uroot -ppasswort shutdown
```

```
shell> mysqladmin -uuser -ppasswort -hhost processlist
```

**mysqldump**

Dieses Programm erzeugt eine Kopie der Datenbank oder Teile davon als ASCII-Datei in vollständiger SQL-Syntax.

- Kann zur Erstellung eines Speicherauszugs einer Datenbank oder einer Sammlung von Datenbanken zu Sicherungszwecken verwendet werden
- Kann zur Übertragung von Daten auf einen anderen SQL-Server verwendet werden
- Der Speicherauszug enthält SQL-Anweisungen zur Erstellung und/oder zum Ausfüllen einer Tabelle.

## Aufruf von mysqldump

```
shell> mysqldump [options] Datenbankname [Tabellen]
```

```
shell> mysqldump [options] --databases Datenbankname1  
[Datenbankname2 Datenbankname3...]
```

```
shell> mysqldump [options] --all-databases
```

**mysqlcheck**

mysqlcheck verwendet die SQL-Anweisungen CHECK TABLE, REPAIR TABLE, ANALYZE TABLE und OPTIMIZE TABLE auf eine für den Benutzer praktische Weise.

Drei Möglichkeiten, mysqlcheck aufzurufen sind:

```
shell> mysqlcheck [options] Datenbankname [Tabellen]
```

```
shell> mysqlcheck [options] --databases Datenbankname1  
[Datenbankname2 Datenbankname3...]
```

```
shell> mysqlcheck [options] -all-databases
```

Das Standardverhalten des Überprüfens von Tabellen (--check) kann geändert werden, indem die Binärdatei umbenannt wird.

=> eine im Vergleich zu anderen Clientprogrammen spezielle Funktion

Namen, die verwendet werden können, um das Standardverhalten von mysqlcheck zu ändern sind:

mysqlrepair Die Standardoption ist --repair.

mysqlanalyze Die Standardoption ist --analyze.

mysqloptimize Die Standardoption ist --optimize.

**mysqlimport**

- stellt eine Befehlszeilenoberfläche für die SQL-Anweisung **LOAD DATA INFILE** bereit.
- Die meisten Optionen für mysqlimport entsprechen direkt Klauseln der LOAD DATA INFILE-Syntax

## Beispiel: Verwendung von LOAD DATA INFILE

```
shell> cat kunden.txt
```

100	Tobias Schmidt	DBA
200	Martin Schultz	Sysadmin
300	Maria Schnieder	Developer

Mit dem Befehl „cat“ wird eine neue Textdatei kunden.txt angelegt und geschrieben, STRG+D gespeichert.



## Beispiel: Verwendung von LOAD DATA INFILE

```
shell> mysql --user=username --password=passwort  
mysql> USE Datenbankname;  
mysql> LOAD DATA LOCAL INFILE '/Pfad/zu/kunden.txt'  
-> INTO TABLE kunden  
-> FIELDS TERMINATED BY '\t'  
-> LINES TERMINATED BY '\n'  
-> (kundenid, name, position);
```

## Aufruf von mysqlimport

```
shell> mysqlimport [options] Datenbankname Textdatei1 [Textdatei2  
Textdatei3 ...]
```

Bei jeder Textdatei, die auf der Befehlszeile angegeben wird, entfernt mysqlimport die Dateinamenserweiterung und bestimmt anhand des Ergebnisses den Namen der Tabelle, in die der Inhalt der Datei importiert werden soll.

=> Dateien mit den Namen kunden.txt, kunden.text und kunden werden alle in eine Tabelle namens kunden importiert.

## Beispiel

```
shell> mysqlimport -u root -ppasswort --local test kunden.txt
```

--local: Liest die Eingabedateien (kunden.txt) lokal vom Clienthost.

**mysqlshow**

- Damit lassen sich Informationen zu den vorhandenen Datenbanken, deren Tabellen oder den Spalten oder Indizes einer Tabelle schnell anzeigen.
- mysqlshow stellt eine Befehlszeilenoberfläche für mehrere SQL-**SHOW**-Anweisungen bereit.

## Aufruf von mysqlshow

```
shell> mysqlshow [options] [Datenbankname [Tabellenname  
[Spaltenname]]]
```

- Wenn keine Datenbank angegeben wird, erscheint eine Liste aller Datenbanknamen.
- Wird keine Tabelle angegeben, dann werden alle passenden Tabellen in der Datenbank angezeigt.
- Wird keine Spalte angegeben, dann werden alle passenden Spalten und Spaltentypen in der Tabelle angezeigt.

## Hilfsprogramme

Hilfsprogramme, die keine Clientverbindung mit dem Server herstellen, sondern MySQL-spezifische Operationen durchführen sind verfügbar.

## Beispiele

### **mysqlbinlog:**

Programm zum Lesen von Anweisungen aus einem Binärlog.

### **myisamchk:**

Wird verwendet, um MyISAM-Tabellen zu beschreiben, überprüfen, optimieren und reparieren.

### **myisamlog:**

Verarbeitet den Inhalt einer MyISAM-Logdatei

## **myisampack:**

Komprimiert MyISAM-Tabellen => erzeugt kleinere schreibgeschützte Tabellen

## **mysqlhotcopy**

Ein Hilfsprogramm, welches schnelle Backups von MyISAM-Tabellen erstellt, während der Server ausgeführt wird.

## **mysql\_zap:**

Damit werden Prozesse terminiert, bei denen eine Übereinstimmung für einen Mustervergleich vorliegt.

## **perror:**

Zeigt die Bedeutung von System- oder MySQL-Fehlercodes an

## **replace:**

Ersetzt Strings in eingegebenem Text