



## Datenbankprogrammierung in Java

Java bietet eine einheitliche **Datenbankschnittstelle** - JDBC. Damit können Sie **so gut wie jede** Datenbank in Ihre Anwendungen **ein binden**. Mit der Heft-CD-ROM erhalten Sie **alles**, was Sie **zum Entwickeln** von Datenbankanwendungen **benötigen**.



# Kaffeespeicher

OLIVER MÜLLER

**A**uf der Heft-CD-ROM liefern wir Ihnen alles mit, was Sie zum Einstieg in die Datenbankprogrammierung benötigen. Neben den Java-Tools, finden Sie auf der Silberscheide die Datenbank MySQL für Linux und Windows. Den passenden JDBC-Treiber liefern wir Ihnen selbstverständlich auch gleich mit.

### ■ Datenspeicher für Pinguine

Die Installation von MySQL unter Linux erfordert wieder etwas Tippgewandtheit. Sie müssen nämlich auf der Shell einige Kommandos eingeben, um das Datenbanksystem zu installieren, zu starten und zu administrieren.

Loggen Sie sich zuerst als Super-User "root" ein. Sollten Sie direkt in der grafischen Oberfläche X11 landen, öffnen Sie ein Shell-Fenster (häufig auch Terminal genannt).

Zuerst müssen Sie eine neue Benutzerkennung und Gruppe für MySQL anlegen. Geben Sie dazu die Befehle

```
groupadd mysql
useradd -g mysql mysql
```

Legen Sie die Heft-CD in das Laufwerk ein und mounten Sie diese durch den Befehl

```
mount /mnt/cdrom
```

bzw. unter SuSE Linux mit

```
mount /cdrom
```

Wechseln Sie in das Verzeichnis /usr/local:

```
cd /usr/local
```

Der nächste Schritt ist das Entpacken des MySQL-Tarballs. Dies erreichen Sie durch das Kommando:

```
tar xzf /mnt/cdrom/MySQL/mysql-3.23.29-gamma-pc-linux-gnu-i686.tar.gz
```

bzw. unter SuSE:

```
tar xzf /cdrom/MySQL/mysql-3.23.29-gamma-pc-linux-gnu-i686.tar.gz
```

Danach können Sie die CD-ROM wieder unmounten und aus dem Laufwerk nehmen:

```
umount /mnt/cdrom
```

bzw. unter SuSE

```
umount /cdrom
```

Nachdem Sie einen Link mit

```
ln -s mysql-3.23.29-gamma-pc-linux-gnu-i686 mysql
```

von mysql-3.23.29-gamma-pc-linux-gnu-i686 nach mysql angelegt haben, wechseln Sie über

```
cd mysql
```

in das Verzeichnis von MySQL. Die "grant table" von MySQL, die die Zugriffsrechte für den Datenbankserver festlegt, erzeugen Sie durch den Befehl

```
scripts/mysql_install_db
```

Diesen Befehl müssen Sie nur ausführen, wenn Sie nicht zuvor schon eine andere MySQL-Distribution installiert hatten.

Zu guter Letzt müssen Sie noch den Besitzer und die Gruppe des MySQL-Verzeichnisses ändern:

```
chown -R mysql:mysql /usr/local/mysql
```

### QUICK INDEX

- ▶ **Datenspeicher für Pinguine**  
MySQL läuft unter Linux.
- ▶ **Datenbank im Fenster**  
MySQL können Sie auch unter Windows einsetzen.
- ▶ **Java an Datenspeicher**  
JDBC lehnt sich vom Konzept sehr an ODBC an.
- ▶ **Der erste Kontakt**  
Damit Sie eine Datenbank verwenden können, müssen Sie eine Verbindung aufbauen.
- ▶ **Nachgefragt**  
Datenbanken fragen Sie über `executeQuery()` ab.
- ▶ **Hoch aktuell**  
Der Datenbankzugriff erfolgt über `executeUpdate()`.



Den Datenbankserver starten Sie durch

```
/usr/local/mysql/bin/safe_mysqld -
-user=mysql &
```



**AWenn Sie MySQL beim Hoch- und Herunterfahren des Systems automatisch starten wollen, finden Sie die passenden Informationen im Kasten "MySQL automatisch".**

Damit Sie nicht immer mühevoll den kompletten Pfad der Tools von MySQL eintippen müssen, nehmen Sie am Besten den Pfad `/usr/local/mysql/bin` in die PATH-Variable auf. Wollen Sie die Programme nur für bestimmte Benutzer zugänglich machen, tragen Sie die folgende Zeile

```
export PATH=$PATH:/usr/local/
mysql/bin
```

in die Datei `~/.bashrc` oder `~/.bashrc_profile` der einzelnen Benutzer ein.

Sollen alle Benutzer auf die MySQL-Dienstprogramme zugreifen können, ist es sinnvoller Sie tragen diese Zeile in die Datei `/etc/profile` ein. Sie müssen dann nur diese eine Datei abändern.

## ■ Datenbank im Fenster

Die Installation unter Windows ist durch den grafischen Assistenten etwas einfacher. Entpacken Sie zunächst das Archiv `mysql-3.23.29a-gamma-win.zip` mit einem ZIP-Packer, wie PkZip oder WinZIP, in ein beliebiges Verzeichnis, zum Beispiel `C:\MS`. Bitte entpacken Sie

dieses ZIP-Archiv nicht in das Verzeichnis `C:\MYSQL`, da dieser Ordner nach Möglichkeit für den installierten Server verwendet werden soll.

Starten Sie danach das Programm `SETUP.EXE` aus diesem Ordner. Es erscheint ein grafischer Installationsassistent, wie Sie ihn auch von anderen Windows-Anwendungen her kennen.

Die Angaben, die Sie in den folgenden Dialogen machen müssen, beschränken sich auf den Zielpfad. Sie müssen lediglich angeben, wo Sie MySQL installieren wollen. Per Default ist hier `C:\mysql` angegeben. Ändern Sie dieses Verzeichnis nach Möglichkeit nicht. Wenn es nicht anders geht und Sie einen anderen Ordner angeben müssen, ist dies auch kein Beinbruch. Sie müssen lediglich beim Start von MySQL diesen anderen Pfad angeben müssen.

Nachdem die Installation durchgeführt ist, können Sie das alte Verzeichnis mit den Setup-Dateien (zum Beispiel `C:\MS`) wieder löschen. Diese Dateien benötigen Sie jetzt nicht mehr.

## Starten unter Windows 9x

MySQL starten Sie unter Windows 95 und 98, indem Sie entweder über in einer MS-DOS-Eingabeoberfläche oder über Programme / Ausführen folgende Befehlszeile eingeben:

```
c:\mysql\bin\mysqld
```

Runlevel 3 und 5 anlegen, damit MySQL in diesen Levels gestartet werden kann. Auch die Lage der Runlevel-Verzeichnisse differiert von Distribution zu Distribution.

Für Red Hat und Co. verwenden Sie im folgenden Befehl die Angaben `/etc/rc.d/rc3.d` und `/etc/rc.d/rc5.d`. Unter Debian setzen Sie stattdessen `/etc/rc3.d` und `/etc/rc5.d` ein. Bei SuSE geben Sie `/sbin/init.d/rc3.d` und `/sbin/init.d/rc5.d` an.

Legen Sie nun zwei Links nach diesem Schema an:

```
ln -s /etc/rc.d/init.d/mysql.server
/etc/rc.d/rc3.d/s15mysql.server
ln -s /etc/rc.d/init.d/mysql.server
/etc/rc.d/rc5.d/s15mysql.server
```

Danach wird MySQL automatisch beim Booten gestartet und beim Shutdown gestoppt.

Wenn Sie ein anderes Verzeichnis als `c:\mysql` bei der Installation angegeben haben, müssen Sie diesen Befehl entsprechend anpassen. Angenommen Sie haben `D:\Programme\MySQL` als Installationspfad angegeben, dann müssen Sie den Datenbankserver durch folgendes Kommando starten:

```
D:\Programme\MySQL\bin\mysqld -
basedir=D:\Programme\MySQL
```

Wollen Sie den Server beenden, verwenden Sie den Befehl

```
c:\mysql\bin\mysqldadmin -u root
shutdown
```



**Aliagt Ihr MySQL-System nicht in C:\mysql müssen Sie den Pfad entsprechend anpassen. Die Option `--basedir <Pfad>` müssen Sie aber nicht angeben.**

Nachdem Sie wissen, welches Kommando MySQL startet, können Sie in Ihren Autostart-Order eine Verknüpfung anlegen. Tragen Sie in dieser neuen Verknüpfung einfach den oben genannten Befehl zum Starten des Servers ein.

Noch ein Tipp am Rande: Der MySQL-Server kommuniziert mit seinen Clients über die Netzwerkprotokoll-Suite TCP/IP. Das heißt konkret, dass Sie TCP/IP auf Ihrem System einrichten müssen - auch wenn Sie an kein Netzwerk angebunden sind. MySQL setzt dabei WinSock2 voraus. Wenn Sie ein altes Windows 95 verwenden, ist womöglich noch nicht diese Version von WinSock enthalten. Laden Sie sich hierzu bei Microsoft den entsprechenden Patch für Windows 95 herunter und installieren Sie ihn. Damit Sie die Dienstprogramme von MySQL direkt ohne Eingabe des gesamten Pfads auf der Eingabeaufforderung verwenden können, tragen Sie das MySQL-bin-Verzeichnis am besten in die PATH-Variable ein. Nehmen Sie folgende Zeile in die Datei `C:\AUTOEXEC.BAT` auf:

```
set PATH=%PATH%;C:\mysql\bin
```

Voraussetzung ist dabei, dass Sie MySQL auch unter `C:\mysql` installiert haben. Ist dies nicht der Fall, müssen Sie die obige Zeile entsprechend abändern.

## Starten unter Windows NT/2000

Bevor Sie MySQL unter Windows NT 4.0 versuchen zu starten, sollten Sie auf jeden Fall sicherstellen, dass mindestens Service Pack 3 installiert ist. Ist dies nicht der Fall wird MySQL auf Ihrem System nicht laufen!

## MYSQL AUTOMATISCH

MySQL können Sie durch ein mitgeliefertes Init-Script automatisch beim Booten und Herunterfahren des Computers starten bzw. beenden. Dieses Script `mysql.server` finden Sie im Verzeichnis `/usr/local/mysql/supportfiles`. Sie müssen es in Ihre Startscripts einbinden und müssen sich danach um nicht mehr weiter kümmern.

Zunächst müssen Sie die Lage Ihrer Init-Scripts herausfinden. Unter Red Hat und hierzu kompatiblen Systemen, finden Sie diese unter `/etc/rc.d/init.d`. Debian/GNU Linux verstaut diese unter `/etc/init.d`. Bei SuSE liegen Sie unter `/sbin/init.d`.

Wechseln Sie in dieses Verzeichnis, und kopieren Sie das MySQL-Startscript:

```
cd /etc/rc.d/init.d
cp /usr/local/mysql/support
files/mysql.server .
```

Als nächstes müssen Sie Links für die



Normalerweise werden (und sollten) Sie MySQL als Service unter Windows NT/2000 starten. Geben Sie hierzu auf der MS-DOS-Eingabeoberfläche bzw. über *Programme/Ausführen* folgenden Befehl ein:

```
C:\mysql\bin\mysqld-nt -install
```

Wenn Sie MySQL in einem anderen Verzeichnis als C:\mysql installiert haben, müssen Sie dieses Kommando anpassen und eine weitere Option anfügen. Angenommen Sie haben MySQL unter D:\MySQL installiert, verwenden Sie dieses Kommando:

```
D:\MySQL\bin\mysqld-nt -basedir=D:\MySQL -install
```

Danach können Sie über die Befehle

```
net start mysql
```

und

```
net stop mysql
```

starten bzw. stoppen.

Sowie Sie den MySQL-Service einmal installiert haben, können Sie ihn auch über den "Services Control Manager" der Systemsteuerung starten und beenden.

Sollen Sie Probleme haben MySQL als Service zu starten, können Sie den Server auch als Standalone-Anwendung verwenden. Verwenden Sie hierzu das selbe Kommando wie für die Installation des Service. Geben Sie nur statt der Option *-install* das Flag *-standalone* an.

Die Dienstprogramme von MySQL zu verwenden und dabei jedesmal auf der Kommandozeile den kompletten Pfad einzutippen, ist nicht gerade komfortabel. Tragen Sie daher das MySQL-bin-Verzeichnis in die PATH-Variable ein.

Unter Windows NT/2000 öffnen Sie hierzu die Systemsteuerung und klicken auf *System*. Wählen Sie dort den Reiter *Umgebung* und hängen Sie an den Wert der PATH-Variablen den Text

```
;C:\mysql\bin
```

an. Sollten Sie MySQL in einem anderen Verzeichnis installiert haben, müssen Sie

die Angabe entsprechend abändern.

## Java an Datenspeicher

Der Zugriff auf Datenbankserver erfolgt in Java über die "Java Database Connectivity", oder kurz JDBC. Bei dieser Datenbankschnittstelle siegte die Pragmatik auch in Java über die Objektorientierung. Anstelle eines objektorientierten Ansatzes setzten die Väter von JDBC darauf, die große Anzahl der vorhandenen SQL-Datenbanksystemen anzubinden.

Das Ergebnis ist ein "Call Level Interface" zum SQL-Server. Die SQL-Befehle werden bei dabei als Strings im Programm verarbeitet und über parametrisierbare Methoden an den Server gesendet.

Dieser wertet den Befehl aus und liefert eine Menge

### LISTING 1

```
import java.sql.*;

public class DemoConnect {
    public static void main(String args[]) {
        Connection con = null;

        try {

            // Treiber laden
            Class.forName("org.gjt.mm.mysql.Driver");

            // Verbindung aufbauen zu
            con = DriverManager.getConnection(
                "jdbc:mysql:localhost:3306/test",
                "der_user",
                "das_passwort");

            // ...
            // Hier folgenden nun
            // die Datenbankoperationen
            // ...

        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            if (con != null) {
                try {
                    // Verbindung schließen
                    con.close();
                } catch (Exception e) { }
            }
        }
    }
}
```

### LISTING 2

```
import java.util.*;
// ...

// Statement-Objekt erzeugen
Statement stmt = con.createStatement();

// Abfrage ausführen
ResultSet rs = stmt.executeQuery("select * from telefon");

// Ergebnisliste ausgeben
System.out.println("Ausgabe:");
while(rs.next()) {
    // die einzelnen Felder in Strings einlesen...
    String Vorname = rs.getString("vorname");
    String Name = rs.getString("name");
    String TelefonNr = rs.getString("telefonnummer");
    // ...und ausgeben
    System.out.println(Vorname + " " + Name + ": " + TelefonNr);
}

stmt.close();
```

## GETCONNECTION()

Methode	Beschreibung
static Connection getConnection (String url);	Öffnet eine Datenbankverbindung zur angegebenen URL.
static Connection getConnection (String url, String user, String password);	Öffnet eine Datenbankverbindung zur angegebenen URL und verwendet als Zugangsdaten den User und das Passwort.
static Connection getConnection (String url, Properties info);	Öffnet eine Datenbankverbindung zur angegebenen URL und übergibt die Parameter für die Verbindung als Properties. Welche Properties hier gesetzt werden können, hängt vom jeweiligen JDBC-Treiber ab.



von Strings als Ergebnis zurück. Durch diese Ergebnisliste "blättert" das Java-Programm durch entsprechende Methoden.

Die sich aufdrängende Ähnlichkeit zu ODBC (Open Database Connectivity) kommt nicht von ungefähr. Das Konzept von ODBC stand für JDBC Modell.

Um einen Datenbankserver per JDBC ansprechen zu können, benötigen Sie einen JDBC-Treiber. Die Treiber unterteilen sich dabei in vier Typen, wie die Bilder zeigen.

**Typ 1:** Diese Kategorie definiert die JDBC-ODBC-Bridge. Der JDBC-Treiber übersetzt die JDBC-Calls in ODBC-Aufrufe. Der JDBC-Treiber ist damit nichts anderes, als eine Fassade, die eine dahinter liegende ODBC-Datenquelle verwendet.

**Typ 2:** Ein Treiber vom Typ 2 verwendet eine datenbankspezifische Schnittstelle in nativem Code (=plattformabhängig). Der Treiber besteht aus Java und plattformabhängigem Code.

**Typ 3:** Treiber vom Typ 3 sind rein in Java implementiert. Der Treiber als solches ist also plattformunabhängig. In Typ 3 ist er jedoch auf eine Middleware (Database Access Server) angewiesen, über den die "echten" Treiber der Datenbanken angesprochen werden.

**Typ 4:** Diese Typkategorie ist schließlich und endlich komplett plattformunabhängig. Ein JDBC-Treiber in reinem Java kommuniziert direkt mit dem Server in dem für das Datenbanksystem erforderlichen Protokoll. Eine Schnittstelle in nativem Code wie in Typ 2 entfällt.

### Der erste Kontakt

Um überhaupt mit JDBC arbeiten zu können, müssen Sie folgende import-Anweisung in Ihr Programm aufnehmen:

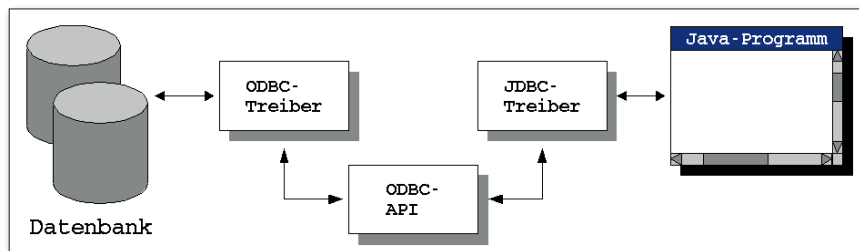
```
import java.sql.*;
```

Damit Sie mit einer Datenbank arbeiten können, müssen Sie eine Verbindung zu ihr herstellen. Dies geschieht in zwei Schritten:

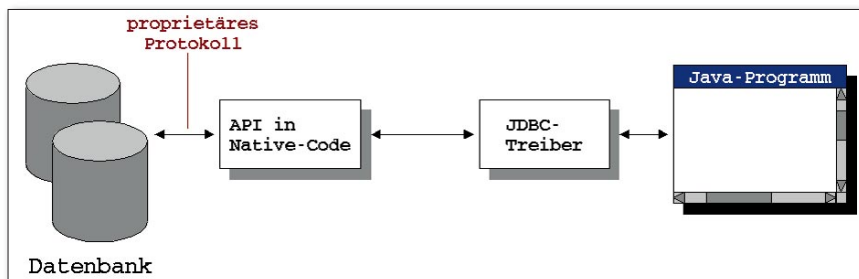
1. Treiber laden.

2. Verbindung aufbauen.

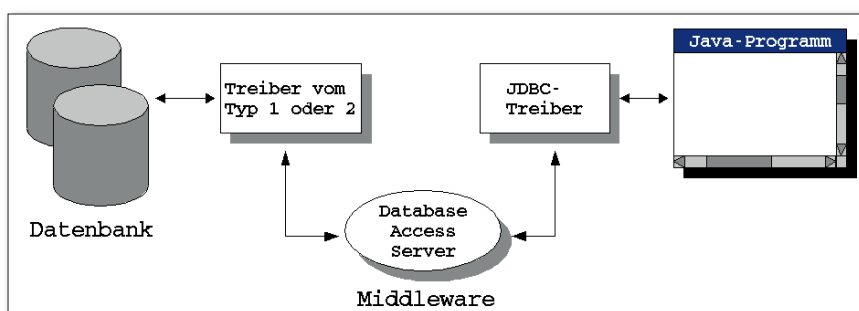
Treiber sind in JDBC/Java selbstverständlich auch Klassen. Um einen Treiber zu laden, verwenden Sie daher die Methode `forName()` der Klasse `Class`.



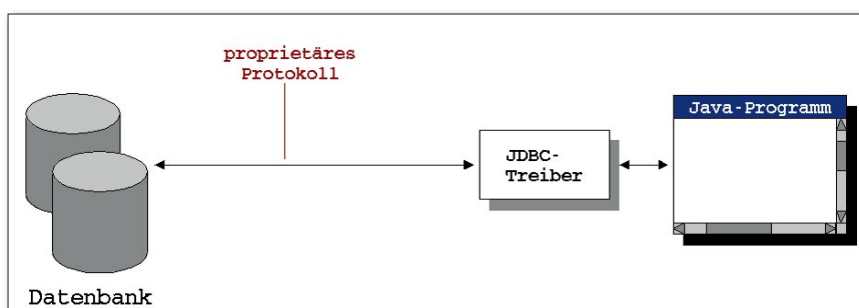
DER JDBC-TREIBER VOM TYP 1 definiert die JDBC-ODBC-Bridge.



DER JDBC-TREIBER VOM TYP 2 verwendet native Code.



DER JDBC-TREIBER VOM TYP 3 sind rein in Java implementiert.



DIE JDBC-TREIBER VOM TYP 3 sind plattformunabhängig.

### WO LAUFEN SIE DENN?

Die Klassen für den JDBC-Treiber von MySQL finden Sie auf der Heft-CD im Verzeichnis `MySQL`. Das komplette Paket ist in dem JAR-Archiv `mm.mysql-2.0.2-bin.jar` verstaut.

Kopieren Sie diese Datei an einen beliebigen Platz in Ihrem Dateisystem, zum Beispiel in das Verzeichnis `/usr/local/mysql-jdbc` oder `C:\mysql-jdbc`. Damit sowohl der Java-Compiler `javac`, als auch der Interpreter `java` bzw. `jre` die Treiber finden, müssen Sie das JAR-Archiv über die Option `-cp` angeben. Angenommen Sie haben das JAR-Archiv in das oben genannte Verzeichnis kopiert. Sie wollen jetzt die Klasse `Adressen.class`, die den JDBC-Treiber verwendet, mit `java` starten. Dann geben Sie als Kommando

```
java -cp /usr/local/mysql
-jdbc/mm.mysql-2.0.2-bin.jar
bzw.
```

```
java -cp C:\mysql-jdbc\mm.
mysql-2.0.2-bin.jar
```

ein



Wie die Treiberklasse heißt, hängt ganz von der Datenbank ab. Um MySQL ansprechen zu können, verwenden Sie die Klasse `org.gjt.mm.mysql.Driver`.

Diese Klasse übergeben Sie der Methode als String als kompletten Klassenpfad:

```
Class.forName("org.gjt.mm.mysql.  
Driver");
```

Nach diesem Aufruf ist der Treiber geladen. Damit Java diese Klasse auch findet, beachten Sie bitte die Hinweise im Kasten "Wo laufen Sie denn?".

**TIPP** Wenn Sie die ODBC-JDBC-Bridge verwenden wollen, verwenden Sie die Treiberklasse `sun.jdbc.odbc.JdbcOdbcDriver`.

Eine Verbindung stellen Sie über die statische Methode(n) `getConnection()` des `DriverManager` her. Die Aufrufformen dieser Methode sind in der gleichnamigen Tabelle aufgeführt. Die Datenbank wird bei diesen Methoden als URL angegeben. Diese hat immer die Form "jdbc:Protokoll:Datenbankpfad". Als Protokoll geben Sie beispielsweise `mysql` für MySQL an oder `odbc` für die ODBC-JDBC-Bridge.

Der Datenbankpfad beginnt immer mit dem Pfad des Server, auf dem die Datenbank im Netzwerk liegt. Für den eigenen Computer ist dies `localhost` gefolgt von dem TCP/IP-Port, auf dem Server läuft (bei MySQL in der Regel 3306). Zum Abschluss des Ganzen folgt

## CRASH-KURS MYSQL-ADMINISTRATION

Das primäre Werkzeug zur Administration eines MySQL-Servers ist `mysqladmin`. Über dieses Kommandozeilentool können Sie zum Beispiel Datenbanken anlegen und löschen oder Passwörter ändern.

**TIPP** Die Kommandoform von `mysqladmin` ist unter Linux und Windows gleich.

Das erste, das Sie nach der Installation von MySQL durchführen sollten, ist das Ändern des "root"-Passwortes (root = Datenbankadministrator). Hierzu rufen Sie das Tool durch folgenden Befehl auf:

```
mysqladmin -u root password  
PASSWORT
```

Statt `PASSWORT` setzen Sie hier ein Passwort Ihrer Wahl ein. Zum Anlegen einer Datenbank verwenden Sie den Befehl

```
mysqladmin -p create DATENBANKNAME
```

Dieser Befehl erzeugt die Datenbank `DATENBANKNAME`.

Bitte beachten Sie, dass sich die MySQL-Versionen unter Windows und Linux bei Datenbank- und Tabellennamen unterscheiden. MySQL bildet Datenbanken und Tabellen auf Dateien des Dateisystems ab. Da Linux - wie jedes Unix-Derivat - zwischen Groß- und Kleinschreibung unterscheidet, Windows jedoch nicht, kommt es hier zu verschiedenen Interpretationen. Die Datenbanken "Telefon" und "telefon" wären unter Win-

dows ein und dieselbe. Unter Linux sind es jedoch verschiedene.

Wollen Sie eine Datenbank wieder löschen, verwenden Sie den Befehl

```
mysqladmin -p drop DATENBANKNAME
```

Den Server-Status können Sie über

```
mysqladmin status
```

abrufen. Die erhalten durch diesen Befehl Informationen über *Uptime*: Zeit in Sekunden, die der Server schon läuft.

*Threads*: Die Anzahl von Threads, die derzeit mit dem Server kommunizieren. *Questions*: Die Anzahl der Abfragen, die seit dem der Server läuft, an diesen geschickt wurden.

*Slow queries*: Die Anzahl an Abfragen, die mehr Zeit benötigt haben, als dies in der Konfiguration festgelegt wurde.

*Opens*: Die Zahl der Tabellen, die seit dem Start des Servers geöffnet wurden. *Flush tables*: Die Anzahl der gelöschten oder neu von der Festplatte gelesenen Tabellen.

*Open tables*: Die Zahl der momentan geöffneten Tabellen.

Eine Aufstellung mit detaillierten Informationen zu den einzelnen Threads erhalten Sie mit

```
mysqladmin processlist
```

Sie sehen in dieser Liste zum Beispiel, wer von welchem Hosts aus, auf Ihre Datenbank zugreift.

## DIE USER-TABELLE

Feld	Beschreibung
Host	Name des/der Hosts, von dem der Benutzer zugreifen darf. Das Zeichen % steht dabei als Platzhalter für beliebig viele Zeichen. Die Angabe <code>test%host</code> , würde auf <code>test22host</code> <code>testzzhost</code> oder <code>test3host</code> passen. Ein einfaches % gestattet den Zugriff von jedem Hosts aus (Achtung: Sicherheitsgefahr!).
User	Die Benutzerkennung.
Password	Das Passwort.
Select_priv	Gibt an, ob der User SELECT-Anweisungen ausführen darf.
Insert_priv	Gibt an, ob der User neue Datensätze durch INSERT einfügen darf.
Update_priv	Sagt aus, ob der Benutzer UPDATE-Befehle ausführen kann.
Delete_priv	Legt fest, ob der User DELETE ausführen darf.
Create_priv	Entscheidet darüber, ob der Benutzer den CREATE-Befehl verwenden kann.
Drop_priv	Gibt an, ob der User DROP ausführen darf.
Reload_priv	Dieses Feld sagt auf, ob der User Zugriffsinformation durch <code>mysqladmin reload</code> einlesen darf.
Shutdown_priv	Entscheidet, ob der User den Server über <code>mysqladmin shutdown</code> herunterfahren darf.
Process_priv	Gibt an, ob der Benutzer Serverprozesse verwalten kann.
File_priv	Sagt aus, ob der Benutzer Dateien mit Befehlen á la <code>SELECT INTO OUTFILE</code> bzw. <code>LOAD DATA INFILE</code> lesen und schreiben darf.
Grant_priv	Entscheidet darüber, ob der User seine Privilegien auf andere Benutzer übertragen darf.
Index_priv	Gibt an, ob der Benutzer Indizes anlegen und löschen darf.
Alter_priv	Sagt aus, ob der User den Befehl <code>ALTER TABLE</code> ausführen darf.



dann noch der Name der Datenbank. Heißt die Datenbank zum Beispiel *test*, können Sie über die URL "jdbc:mysql://localhost:3306/test" eine Verbindung zu ihr aufbauen.

Als Resultat erhalten Sie von der Methode *getConnection()* ein Objekt der Klasse *Connection*, das die Verbindung zum Server darstellt. Einen Rahmencode für den Aufbau einer Verbindung sehen Sie in Listing 1.

### ■ Nachgefragt

Zum Abfragen einer Datenbank benötigen Sie zunächst eine Instanz der Klasse *Statement*. Dieses Objekt erhalten Sie über die Methode *createStatement()* eines *Connection*-Objekts.

Dieses *Statement*-Objekt erhält über die Methode *executeQuery()* eine SQL-

Anweisung – genauer ein *SELECT*-Statement – als String. Diesen SQL-Befehl sendet das *Statement*-Objekt weiter an den Datenbankserver. Dieser führt es aus und liefert seine Ergebnisse an das *Statement*-Objekt. Dieses wiederum erzeugt eine Ergebnismenge der Klasse *ResultSet*.

Das zurückgegebene *ResultSet*-Objekt kann durch die Methode *next()* die Ergebnismenge durchblättern. Diese Methode gibt solange *true* zurück, bis das Ende der gefundenen Datensätze erreicht wurde.

Auf den aktuellen Datensatz des *ResultSet*-Objekts können Sie die *get*-Methoden des *ResultSet*-Objekts zugreifen. Eine Übersicht der *get*-Methoden sehen Sie in der Tabelle "get-Methoden". Diese Methoden erwarten jeweils

als String den Namen des Datenbankfeldes und liefern den Inhalt dieses Feldes konvertiert in einen Java-Typ bzw. ein Objekt zurück. Angenommen Sie haben in einer Datenbank eine Tabelle mit dem Namen *telefon* gespeichert. Diese

enthält die Felder *vorname*, *name* und *telefonnummer*. Den Zugriff auf diese Tabelle zeigt Ihnen der Code-Abschnitt in Listing 2. Diesen Code-Teil setzen Sie – bis auf das *import*-Statement – in Listing 1 ein, wo der Kommentar als Platzhalter für die Datenbankoperationen angegeben ist. Die *import*-Anweisung setzen Sie selbstverständlich an den Anfang des Programmlistings.

Das Programm erzeugt erst ein *Statement*-Objekt *stmt* und führt dann eine Abfrage per *SELECT*-Befehl durch. Die Ergebnisse werden im *Result*-Objekt *rs* gespeichert. Danach tritt das Programm in eine Schleife ein, die solange ausgeführt wird, bis das *Result*-Objekt *rs* nicht mehr mit *next()* "weitergeblättert" werden kann.

Die Feldinhalte des jeweils aktuellen Datensatzes der Ergebnisliste werden über *getString()* in Strings eingelesen. Diese werden zum Schluss ausgegeben und eine neue Runde der Schleife beginnt mit dem nächsten Datensatz.

### ■ Hoch aktuell

Wenn Sie andere SQL-Befehle als *SELECT* einsetzen wollen, benötigen Sie zunächst auch ein *Statement*-Objekt. Dieses holen wieder über die Methode *createStatement()* des *Connection*-Objekts. Um nun einen SQL-Befehl wie *INSERT INTO* oder *CREATE TABLE* an das Datenbanksystem zu schicken, verwenden Sie die Methode *executeUpdate()* des *Statement*-Objekts. Diese Methode erwartet einen String mit dem SQL-Kommando als Argument. Ein Beispiel für *executeUpdate()* sehen Sie in Listing 3. Es schreibt die Telefonnummer von "Donald Duck" mit dem Befehl *INSERT INTO* in die Tabelle *telefon*.

✓ UR

#### LISTING 3

```
import java.util.*;
// ...

// Statement-Objekt holen
Statement stmt = con.createStatement();

// SQL-Befehl ausführen
stmt.executeUpdate("insert into telefon "
    + " (vorname, name, telefonnummer) "
    + " values('Donald', 'Duck', '{00 00} 0 00 00')");
```

#### BENUTZER VERWALTEN IN MYSQL

Die Benutzerverwaltung in MySQL erfolgt nicht ganz so komfortabel, wie in anderen Systemen. Sie müssen hier eine Datenbanktabelle bearbeiten. Diese Tabelle heißt *user* und befindet sich in der Datenbank *mysql*, die sämtliche Verwaltungsinformationen speichert.

Um einen neuen User anzulegen, müssen Sie für diesen einen Datensatz in der Tabelle *user* erzeugen. Dies können Sie durch den SQL-Interpreter *mysql* bewerkstelligen. Dieses Tool verlangt als

Kommandozeilenargument den Namen der Datenbank (hier *mysql*). Über *-u* und *-p* können Sie Benutzername (*root*) und das passende Passwort angeben.

Die Benutzertabelle bearbeiten Sie mit den herkömmlichen SQL-Befehlen, beispielsweise wie *INSERT INTO*. Den Aufbau von *user* verrät Ihnen die Tabelle "Die *user*-Tabelle". Die *priv*-Einträge dieser Tabelle erlauben entweder ein 'Y' oder ein 'N'.

#### GET-METHODEN

Methode	Rückgabewert	Beschreibung
<i>getBoolean(String Feld)</i>	<i>boolean</i>	Der Wert des Datenbankfeldes wird als booleschen Wahrheitswert geliefert.
<i>getByte(String Feld)</i>	<i>byte</i>	Liefert den Wert des Datenbankfeldes als Byte.
<i>getBytes(String Feld)</i>	<i>byte[]</i>	Gibt den Inhalt des Datenbankfeldes als Byte-Array zurück.
<i>getDate(String Feld)</i>	<i>Date</i>	Liefert den Wert des Datenbankfeldes als Date-Objekt.
<i>getDouble(String Feld)</i>	<i>double</i>	Liefert das Datenbankfeld als Fließkommawert vom Typ <i>double</i> zurück.
<i>getFloat(String Feld)</i>	<i>float</i>	Gibt den Wert des Datenbankfeldes als Fließkommawert vom Typ <i>float</i> zurück.
<i>getInt(String Feld)</i>	<i>int</i>	Das Datenbankfeld wird als Integer zurückgegeben.
<i>getLong(String Feld)</i>	<i>long</i>	Der Inhalt des Feldes wird als <i>long</i> geliefert.
<i>getShort(String Feld)</i>	<i>short</i>	Liefert den Inhalt des Feldes als <i>short</i> zurück.
<i>getString(String Feld)</i>	<i>String</i>	Der Inhalt des Datenbankfeldes wird in einen String konvertiert und zurückgegeben.
<i>getTime(String Feld)</i>	<i>Time</i>	Liefert den Wert des Datenbankfeldes als Time-Objekt.
<i>getTimestamp(String Feld)</i>	<i>Timestamp</i>	Der Wert des angegebenen Feldes wird als Timestamp-Instanz zurückgegeben.