

9 MYSQL.....	3
9.1 EIGENSCHAFTEN VON MYSQL.....	4
9.2 INSTALLATION/BASISKONFIGURATION/ANWENDUNG.....	11
9.2.1 MySQL Distribution.....	14
9.2.2 Installation der Binär-Distribution.....	15
9.2.3 Installation der Source-Distribution.....	16
9.2.4 Basiskonfiguration.....	22
9.2.5 Benutzung des Standard Client mysql.....	28
9.2.6 Hilfsprogramme der MySQL-Distribution.....	32
mysql als Kommandozeilen-Filter.....	32
mysqladmin.....	34
mysqldump.....	35
mysqlimport.....	36
Weitere Hilfsprogramme.....	36
9.2.7 Webmin.....	37
9.2.8 Web-Browser als Client: phpMyAdmin.....	47
9.2.9 Grafische Clients unter Linux.....	75
T.c.X. MySQLGUI.....	76
KSql (früher KMySql).....	80
9.2.10 Microsoft-Windows-Programme als MySQL-Client.....	93
9.3 BENUTZERVERWALTUNG.....	98
9.4 DATENSICHERUNG UND WARTUNG.....	107

9 MySQL

Warum das DBMS MySQL so heißt wie es heißt? Das ist nicht genau überliefert – aber die Tochter von Michael Widenius heißt My (das ist nichts Ungewöhnliches für schwedisch sprechende Finninnen). Und Michael Widenius ist einer der Hauptentwickler von MySQL. Die offizielle Aussprache von MySQL ist übrigens: „Mei Es Kju Ell“ (und nicht etwa „Mei-sikuell“). So steht’s im Handbuch!

MySQL ist sicher keine Einsteiger-Spielzeug-Datenbank. Ihre Verwendung in vielen professionellen Projekten beweist ihre Praxistauglichkeit auch in ernsthaften Anwendungen. MySQL ist aber deutlich weniger komplex als z. B. Oracle und daher auch deutlich schneller zu erlernen. Da MySQL unter Linux in der Regel ohne Entrichtung von Lizenzgebühren verwendet werden kann, stellen sich dem Datenbankanfänger auch keine ökonomischen Hindernisse. Aus diesen Gründen eignet sich MySQL in besonderer Weise für Einsteiger. Dieses Kapitel möchte dem Rechnung tragen, indem es – soweit für den Betrieb der Datenbank notwendig – auch die erforderlichen Aktivitäten auf Betriebssystemebene oder z. B. bei der Konfiguration des Apache Web Servers erläutert. In Kapitel 9.1 besprechen wir zunächst überblicksartig und bezüglich der Eignung von MySQL für unterschiedliche Aufgaben die Eigenschaften dieser DBMS. Es folgt eine ausführliche und detaillierte Darstellung der Installation, Konfiguration und Anwendung von MySQL und seinen Komponenten sowie einiger weiterer Tools (Kapitel 9.2). Dabei wird wieder auch auf die Probleme eingegangen, die sich demjenigen stellen, dessen MySQL-Datenbank von einem der großen Internet Service Provider gehostet wird, und daher auf Betriebssystemebene nicht auf die Datenbank zugreifen kann. In einem eigenen Unterkapitel (9.3) wird ferner ausführlich auf die Benutzerverwaltung von MySQL eingegangen (ein größerer Teil des Traffics in den MySQL-Mailinglisten beschäftigt sich mit Au-

thentizierungsproblemen). Abschließend wird auf die Vermeidung von Datenverlusten durch Datensicherung sowie auf die Reparatur beschädigter Tabellen eingegangen (Kapitel 9.4).

9.1 Eigenschaften von MySQL

MySQL hat gegenüber anderen Datenbanken eine Reihe von Vorzügen, die im folgenden kurz dargelegt werden sollen.

MySQL ist eine bekannte, weit verbreitete Datenbank

Ogleich dies für sich genommen nicht eigentlich ein Qualitätsbeweis ist (wie die Popularität mancher Betriebssysteme beweist), so hat es doch auch seine Vorteile. So finden sich eine Vielzahl von Webressourcen, die eine wertvolle Informationsquelle darstellen. Da MySQL auch im universitären Umfeld (auch in der Ausbildung) gerne eingesetzt wird, finden sich zunehmend IT-Fachleute, denen dieses DMBS sehr vertraut ist.

MySQL hat eine rasante Entwicklung hinter sich. Ursprünglich suchten die Leute von T.c.X. DataKonsult AB (<http://www.tcx.se>) nur einen Ersatz für einen einfachen, in BASIC geschriebenen Datenbank-Reportgenerator, mit dem sie bis dahin gearbeitet hatten. Es erschien ihnen zunächst mSQL geeignet (zu mSQL siehe Kapitel 12); bald stellte sich aber heraus, daß mSQL den Erfordernissen von T.c.X. (schneller Zugriff auf große Datenbestände) nicht genügte. So entschloß man sich zur Eigenentwicklung MySQL. Deren API wurde auf der Grundlage derjenigen von mSQL entwickelt; dies erlaubte eine relativ leichte Portierung von Anwendungen, die für die mSQL API verfügbar waren. Mitte 1996 wurde die Version 1.0 von MySQL einer beschränkten Öffentlichkeit vorgestellt (vier Personen). Diese waren der Auffassung, daß MySQL nicht nur für T.c.X. nützlich sei (eine richtige Einschätzung, wie sich herausstellte). Man

stellte daher noch im gleichen Jahr die erste Source-Distribution (MySQL 3.11.1) einer breiteren Öffentlichkeit zur Verfügung.

Am heutigen Tag ist die Versionszählung bei 3.23.14 angelangt, und MySQL hat sich zu einer der beliebtesten Datenbanken unter Linuxern und hier insbesondere unter Webadministratorinnen und -toren gemausert. Die Suchmaschine Fireball (<http://www.fireball.de>) findet im deutschsprachigen Web für das Stichwort MySQL immerhin knapp 9.000 Treffer. Und wenn sich auch für den Datenbankengiganten Oracle mit knapp 20.000 noch deutlich mehr Suchergebnisse finden, so sind es doch für andere Datenbanken deutlich weniger (Informix: gut 6.000; Sybase: knapp 4.000 Treffer). Die Popularität von MySQL läßt sich auch an der Tatsache erkennen, daß es für die Heimatseite vom MySQL (<http://www.mysql.com>) weltweit inzwischen über 50 Mirrors gibt. Jede deutsche Computerzeitung, die etwas auf sich hält, hat inzwischen über MySQL (häufig in Verbindung mit PHP) berichtet, die c't hat gar den Begriff LAMP-System geprägt. Hiermit ist bereits der nächste Punkt angesprochen:

MySQL eignet sich optimal als Backend fürs Web

Mit der Kombination des Betriebssystems Linux, des Webserver Apache, dem Datenbanksystem MySQL und der Middleware PHP (LAMP-System) lassen sich kostengünstige (softwareseitig in der Regel umsonst) aber dennoch leistungsstarke internet-basierte Datenbankapplikationen aufbauen. Mit der rasanten Ausweitung des Zugangs zum Internet in breite und auch in technisch nicht interessierte Bevölkerungskreise gewinnen interaktive Websites, E-Commerce usw. und damit Datenbanken mit WWW-Schnittstelle immer mehr an Bedeutung. Und dies betrifft keineswegs nur große Konzerne wie Bertelsmann, Buch.de oder Beate Uhse. Auch der ambitionierte Homepagebastler sowie die halb- und vollprofessionelle Freelancerin haben zur Erstellung kleinerer und mittlerer Websites wie

<http://www.dem-boris-seine-site.de> oder auch durchaus ernsthafter und/oder kommerzieller Anwendungen (Schallplattenbörsen, Antiquariate, Webshops usw.) Bedarf an einfach zu bedienenden Web-Datenbankapplikationen.

Wenn man sich die mindestens einmal in der Woche allen gängigen deutschen Tageszeitungen beiliegenden Prospekte großer ISPs anschaut (Puretec, Strato), zeigt sich, daß die „Profi-“ bzw. „Premium“-Pakete dieser Angebote u. a. auch damit werben, daß ein Zugriff auf MySQL möglich sei (womit wir nicht sagen wollen, daß diese Pakete wirklich „profi“ oder „premium“ sind und ein Zugriff auf die beworbene Datenbank bei diesen Anbietern *tatsächlich* möglich ist: glaubt man den diesbezüglichen Verzweiflungsmails in der deutschen PHP-Mailingliste war dies zumindest zeit- und teilweise nicht der Fall). Auch eine Vielzahl kleinerer und mittlerer Internet Service Provider bieten inzwischen MySQL (und PHP) Unterstützung an (siehe hierzu die – keinesfalls vollständige weil noch nicht fertiggestellte – (ebenfalls unter MySQL laufende) Providerdatenbank der deutschsprachigen PHP-Portalsite PHP-Center (<http://www.php-center.de/provider/>).

Daß sich das Gespann PHP/MySQL durchaus auch für größere Internetprojekte eignet, zeigt unter anderem z. B. die Webpräsenz der Expo (<http://www.expo2000.de/>, der weltweite Ticketverkauf für die Expo 2000 ist mit MySQL und TCL/TK implementiert), sowie die in der Linux und Open Source Gemeinde wohlbekanntesten Websites <http://slashdot.org/>, <http://freshmeat.net/> und <http://www.linux.com/>. Auch im universitären Bereich wird in jüngster Zeit vermehrt auf MySQL zurückgegriffen: So verwendet z. B. die Fernuniversität Hagen für "WebAssign" (eine Eigenentwicklung zur Durchführung des internet-basierten Übungsbetriebs) MySQL (<http://niobe.fernuni-hagen.de/WebAssign/>). Und auch das Virtus Projekt (Virtuelle Universität; Einsatz neuer Medien in der universitären Lehre) setzt für seine

datenbankgestützte Lernplattform ILIAS MySQL ein (<http://www.virtus.uni-koeln.de/>).

Mit MySQL lassen sich auch vergleichsweise große Datenbestände verwalten. Die Leute von T.c.X. behaupten, daß sie MySQL in einer Umgebung mit mehr als 40 Datenbanken einsetzen, die zusammen mehr als 10.000 Tabellen enthalten, von denen mehr als 500 mehr als 7 Millionen Reihen beinhalten. Insgesamt mehr als 100 Gigabytes Daten. Das ist sicherlich deutlich mehr, als in einem kleineren bis mittleren Web-Shop anfallen.

MySQL ist ein extrem schnelles DBMS

MySQL ist äußerst stabil und auch bei großen Datenmengen extrem schnell und effizient. MySQL verwaltet seine Indizes - wie andere Datenbanken auch - in Baumstrukturen (B-trees). Auf diese Datenstrukturen kann mit logarithmischer Komplexität zugegriffen werden, d. h. es sind nur sehr wenige Zugriffe notwendig, um einen beliebigen Zieldatensatz über den Index zu finden. Köhntopp schätzt, daß zum Finden eines Datensatzes aus tausend Datensätzen maximal 3, aus einer Million Datensätzen maximal 5 und aus einer Milliarde Datensätzen maximal 7 Vergleiche und Plattenzugriffe erforderlich sind ([kor00]). Auf Geschwindigkeitsoptimierung haben die Entwickler auch sonst großen Wert gelegt: Traut man den Benchmarks der MySQL-Autoren, so ist MySQL im Vergleich mit Oracle, Sybase, Informix, PostgreSQL, IBM DB2, SOLID, EMPRESS, Microsoft SQL Server, Adabas sowie Access 2000 das mit Abstand schnellste Datenbanksystem (<http://www.mysql.com/benchmark.html>). Ein Teil dieser Performance wird (positiv gewendet) durch eine Beschränkung „aufs Wesentliche“ erreicht.

MySQL ist auch für kommerzielle Anwendungen unter Linux umsonst

Die Lizenzbedingungen von MySQL sind sehr moderat. Für Nutzer von Nicht-Microsoft-Betriebssystemen fallen im Regelfall

keine Lizenzgebühren an. Dies gilt auch bei kommerzieller Verwendung und insbesondere auch bei einer Verwendung als Webdatenbank (auch in diesem Punkt unterscheidet MySQL sich positiv von mSQL). Lizenzgebühren sind nur zu zahlen, wenn man MySQL als Teil eines anderen Produkts verkaufen will, man Clients für die Installation und Wartung des MySQL-Servers Rechnungen ausstellen möchte, oder – wenn man MySQL unter Windows 95, 98, NT oder Windows 2000 nutzen möchte. Diese Lizenzgebühren werden fällig, weil es sich bei diesen Betriebssystemen um hochkommerzielle Systeme handelt und die Entwicklung für diese Plattform sehr aufwendig ist und von den Entwicklern als unangenehm empfunden wird; insoweit stellen die Lizenzgebühren eigentlich mehr ein Schmerzensgeld dar.

Features von MySQL:

- Client/Server Architektur. Multi-User.
- MySQL ist vollständig Multi-Threaded. Durch die Nutzung von Kernel Threads können auf einfache Weise – sofern vorhanden – mehrere CPUs genutzt werden.
- Es stehen eine Fülle von Programmierschnittstellen zur Verfügung: Neben C und C++ werden – ebenfalls als Freeware – Java, Perl, Python, PHP und TCL APIs angeboten.
- Es stehen sowohl freie ODBC als auch JDBC Treiber zur Verfügung.
- MySQL läuft auf sehr vielen verschiedenen Plattformen: Selbstverständlich unter Linux aber auch unter AIX, BSDI, DEC UNIX, FreeBSD, HP-UX, MacOS, NetBSD, OpenBSD, OS/2, SGI Irix, Solaris und SunOS, SCO und Tru64 Unix. Ach ja, und unter Windows 95, 98 und NT natürlich auch.

- Das Datenformat der Tabellen ist plattformunabhängig; d. h. Datenbestände können problemlos zwischen verschiedenen Maschinen unter verschiedenen Betriebssystemen ausgetauscht werden.
- Es werden sehr viele verschiedene Spaltentypen unterstützt (einschließlich aller ANSI 92 und aller ODBC 2.50 Typen): signed/unsigned integer-Typen (1, 2, 3, 4 und 8 Bytes lang), FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM.
- MySQL unterstützt fast alle ODBC 3.0 und SQL ANSI92 Funktionen.
- Die SQL GROUP BY und ORDER BY Klauseln sind vollständig implementiert; außerdem werden die GROUP-Funktionen COUNT, AVG, STD, SUM, MAX und MIN unterstützt.
- Es steht eine flexible und sichere Benutzerverwaltung mit host-basierter Verifikation zur Verfügung. Leider ist diese auch entsprechend kompliziert geraten.
- Es können je Tabelle bis zu 16 Indizes erstellt werden. Jeder Index kann aus bis zu 16 Spalten oder Teilen einer Spalte bestehen.
- Auch große Datenbanken mit mehr als 50.000.000 Datensätzen können zuverlässig gehandhabt werden.
- Allen Spalten können Vorgabewerte zugewiesen werden. Wenn Sie mit einem Insert-Statement nur einem Teil der Spalten einen Wert zuweisen, werden die anderen automatisch auf die Vorgabewerte gesetzt.
- Es können Tabellen verschiedener Datenbanken in einer Abfrage gemischt werden.

- LEFT OUTER JOINS werden sowohl in ANSI SQL als auch in ODBC-Syntax verarbeitet.
- Der ISO-8859-1 Latin1 Zeichensatz wird vollständig unterstützt; alle Daten werden entsprechend gespeichert (und bei Bedarf sortiert!).
- Fehlermeldungen des Servers können in verschiedenen Sprachen (und auch in Deutsch) ausgegeben werden.
- Zur Abfrageoptimierungen kann der EXPLAIN-Befehl genutzt werden. Mit ihm läßt sich feststellen, in welcher Weise der Optimizer eine Query auflöst.
- Es stehen eine Vielzahl von Utilities und freier Software (z. B. Clients) zur Verfügung.

Beschränkungen von MySQL.

Aber MySQL hat nicht nur Vorteile, sondern – wie oben bereits angedeutet – auch Beschränkungen. MySQL unterstützt in etwa den Funktionsumfang von SQL92. Es fehlen allerdings „Views“, „Stored Procedures“ und „Triggers“. Nicht implementiert sind derzeit noch Transaktionen (es existiert kein Commit/Rollback-Mechanismus) und Foreign Keys (die referentielle Integrität kann nicht erzwungen werden). In vielen Anwendungen sind diese Beschränkungen aber weniger tragisch, als man zunächst vielleicht vermuten würde. So bedeutet das Fehlen von Transaktionen keinesfalls, daß der simultane Zugriff vieler Nutzer nicht möglich wäre. Im Gegenteil erhält jeder MySQL Client seinen eigenen Thread in den MySQL Server. Auf diese Weise können verschiedene Nutzer zur gleichen Zeit auf die gleiche Tabelle zugreifen. Alle MySQL Operationen sind atomar: kein anderer Nutzer kann das Ergebnis einer laufenden Datenbankabfrage verändern. MySQL gewährleistet, daß niemand in ein spezifisches, laufendes Update eingreifen kann. Im Abschnitt XX

werden die Beschränkungen von MySQL sowie mögliche Lösungen und Workarounds im Detail diskutiert.

Typische Einsatzgebiete von MySQL

Für welche Bedürfnisse und Anforderungen eignet sich MySQL und wann sollten Sie lieber auf ein anderes DBMS ausweichen? MySQL zeichnet sich vor allem durch eine sehr hohe Geschwindigkeit auch bei großen Datenbeständen, durch einen geringen Speicherverbrauch sowie durch äußerste Stabilität aus. MySQL eignet sich hervorragend als Datenbank fürs Web und interagiert schnell und zuverlässig mit dem Apache Webserver (via PHP). Server wie Clients können in der Regel kostenfrei verwendet werden. Probleme bekommen Sie mit MySQL immer dann, wenn Ihre Anwendung zwingend auf referentielle Integrität angewiesen ist (wegen des Fehlens der foreign key Prüfungen). Probleme können aufgrund der eingeschränkten Transaktionsverarbeitung auch entstehen, wenn schreibend auf mehr als eine Tabelle gleichzeitig zugegriffen werden soll. MySQL eignet sich für alle Modelle, auf die überwiegend lesend zugegriffen wird (wie bei den meisten Webanwendungen) oder/und bei denen Schreibzugriffe überwiegend einzelne Tabellen verändern. Für Modelle mit sehr vielen Schreibzugriffen sowie für Modelle mit mehr als zwei Schreibzugriffen, die mehr als eine Tabelle gleichzeitig aktualisieren, ist MySQL hingegen nicht optimal geeignet.

9.2 Installation/Basiskonfiguration/Anwendung

Die Installation von Software unter Linux hat sich in den letzten Jahren deutlich vereinfacht. Die großen Linux-Distributionen (und die kleinen auch) liefern auf ihren Webservern und den CDs häufig angepasste und vorcompilierte Archivdateien die

sich sehr einfach installieren lassen. Sehr verbreitet sind Pakete im Format des `Redhat Packet Managers` (*.rpm). Der Paketmanager von Redhat `rpm` (er ist frei unter GPL verfügbar) hat sich als Quasistandard durchgesetzt und wird nicht nur in der Redhat-Distribution sondern zum Beispiel auch von SuSE, Caldera und DLD genutzt. Leider ist die Linux-Welt deswegen nicht in jedem Fall einfacher geworden, denn die meisten Distributionen verwenden zwar das selbe Paketformat, der Austausch von Paketen zwischen unterschiedlichen Distributionen ist aber nach wie vor schwierig.

Dies liegt zum einen an unterschiedlichen Vorkonfigurationen und voneinander abweichenden Installationspfaden und zum anderen an der Verwaltung der Abhängigkeitsinformationen (welche Libraries werden benötigt usw.), für die es keinen Standard gibt. Wenn Sie sich auf dem Webserver von T.c.X. anschauen (<http://www.mysql.com>) werden Sie die Datenbanksoftware auch als rpm-Paket vorfinden. Sollten Sie Redhat-Linux verwenden, können Sie dieses Paket problemlos verwenden und mit

```
rpm -install paketname.rpm
```

installieren. Sie sollten dieses Paket hingegen *keinesfalls* verwenden, wenn Sie z. B. SuSE-Linux oder eine andere Linux-Distribution installiert haben. SuSE liefert ein eigenes MySQL-RPM-Paket (sowohl auf CD als auch auf dem Webserver <http://www.suse.de> verfügbar), das Sie mit `YAST` installieren können. Im folgenden soll jedoch die Installation mit Hilfe distributionsunabhängiger Tar-Archive besprochen werden. Für MySQL sind sowohl Binär- als auch Source-Archive verfügbar. Im Abschnitt 9.1.1 besprechen wir kurz die Installation der Binär- und ausführlich die der Source-Distribution. Die Installation einer Binär-Distribution ist in der Regel zwar einfacher als die der Source-Distribution, dafür können Sie nur mit einer selbstcompilierten

Version z. B. die Möglichkeiten von User Defined Functions (UDFs) nutzen.

Als Referenzsystem dient uns ein auf einem Pentium III/500 mit 256Mbyte Arbeitsspeicher installiertes SuSE 6.1 System. Häufig treten bei umfangreichen Installationen distributionsabhängige Besonderheiten und Probleme auf. Diese werden hier nicht besprochen (soweit sie nicht das Referenzsystem SuSE 6.1 betreffen). Sollten sich bei der Installation unerwartete Probleme ergeben, lohnt fast immer ein Blick auf die Homepage der jeweiligen Linux-Distribution. SuSE z. B. pflegt eine ausgezeichnete Support-Datenbank, die zu vielen Problemen Hilfestellung geben kann.

In gesonderten Abschnitten besprechen wir die Installation und Verwendung einiger Zusatzpakete, die wir für so nützlich halten, daß wir Ihnen ihre Anwendung empfehlen und die Installation und den Umgang mit ihnen erläutern wollen. Dies betrifft das Paket Webmin (Abschnitt 9.2.2), das Ihnen nicht nur die Administration von MySQL sondern ihres gesamten Linuxsystems über eine Browser-Interface ermöglicht, das Paket PHPMyAdmin (Abschnitt 9.2.3), ein in PHP geschriebener, webbasierter MySQL-Client, der sich insbesondere (aber nicht nur) zur Administration von MySQL-Datenbanken eignet, die von ISPs gehostet werden, die Ihnen keinen direkten Zugriff auf das System via `telnet` ermöglichen und auch den MySQL-Port ihres Systems dicht gemacht haben (und sie daher auf Ihre Datenbank nur noch über das Web zugreifen können).

Ferner gehen wir kurz auf die Installation und Nutzung zweier graphischer Clients für KDE unter Linux ein (Abschnitt 9.2.4). Zuletzt möchten wir Ihnen noch zeigen, wie Sie von Windows aus mit der Frontend-Datenbank Access von Microsoft oder mit dem Tabellenkalkulationsprogramm Microsoft Excel Ihren MySQL-Server unter Linux benutzen und administrieren können. Hierzu muß der MySQL-ODBC-Treiber unter Windows installiert

und konfiguriert werden (Abschnitt 9.2.5). Diese Möglichkeit (und deshalb wird sie hier auch beschrieben) kann manchmal ein ausschlaggebendes Argument gegenüber Kunden oder Chefs sein, die die Leistungsfähigkeit von Software unter Linux oftmals noch unter- und die Leistungsfähigkeit von Produkten der Firma Microsoft überschätzen. Der Autor dieser Zeilen betreibt im Hauptberuf empirische Sozialforschung; aus diesem Grund werden Ihnen in diesem Abschnitt ebenfalls seine Erkenntnisse über Möglichkeiten aufgenötigt, mit Hilfe von Statistikprogrammen wie SPSS auf SQL-Datenbanken (ebenfalls via ODBC) zuzugreifen.

9.2.1 MySQL Distribution

Die Versionsbezeichnungen der MySQL-Distributionen geben einige Hinweise auf den Stand der Entwicklung. Während diese Zeilen geschrieben werden, hat die aktuellste MySQL-Version die Nummer 3.23.13. Solange die erste Ziffer (3) sich nicht ändert, ändert sich auch nicht das Dateiformat. Ändert sich bei einem Update diese Ziffer nicht, können alle bisher bestehenden Datenbanken ohne zusätzliche Maßnahmen übernommen werden (Die Versionen 4.x.x werden allerdings komfortable Tools zur Konvertierung alter Datenbanken enthalten). Die zweite Ziffer gibt den Release Level an. Es stehen immer mindestens zwei Release Level zum Download zur Verfügung (derzeit 3.22 und 3.23). Die dritte Ziffer gibt die Versionsnummer der Release an (z. B. 3.23.13). Die vorletzte Release ist jeweils die vollständig ausgetestete, stabile Version von MySQL. In aller Regel werden Sie aus dieser Release die letzte Version verwenden (also derzeit 3.22.32).

In die aktuelle Release werden regelmäßig neue Funktionen eingepflegt, sie kann daher nicht die gleiche Stabilität haben, die die Vorgänger-Release hat. Die aktuelle Release verfügt

darüberhinaus über einen Stabilitätslevel. Die Release 3.23 ist in ihrer Version 13 noch `alpha`. Eine Release im alpha-Stadium enthält noch ungetesteten, neuen Code. Release 3.23 wird in Kürze aufgewertet und erhält dann den Stabilitätslevel `beta`.

Hinsichtlich der Stabilität können die Versionsbezeichnungen in MySQL getrost als konservativ gelten. Jede MySQL-Version hat umfangreiche Tests durchlaufen, bevor sie zum Download freigegeben wird. T.c.X. verwendet stets die aktuellste Version in ihrer Produktionsumgebung, so daß gravierende Mängel den Entwicklern unverzüglich rückgemeldet werden. Die Release 3.23 ist inzwischen auch sonst weithin im Einsatz. Größere Probleme sind in letzter Zeit nicht bekannt geworden.

Sowohl die Installation der MySQL-Source- als auch die der Binärdistribution gelingt in der Regel problemlos. Die Installation der Binärdistribution wird hier nur kurz angesprochen. Ausführlich werden wir hingegen auf die Installation der Source-Distribution eingehen. Die Basiskonfiguration und Einrichtung des Datenbankservers wird im Anschluß daran erörtert.

9.2.2 Installation der Binär-Distribution

Nachdem Sie sich die erforderliche Archivdatei verschafft haben (Download von einem der MySQL-Mirror bzw. Kopie von CD) kopieren sie diese in das Verzeichnis unterhalb des Ziel-Verzeichnisses (wenn Sie – wie üblich – die Binärdistribution in `/usr/local/mysql` installieren wollen, kopieren Sie das Archiv nach `/usr/local`. Anschließend muß es entpackt und end-tart werden. Zu guter Letzt setzen Sie noch einen Symbolic Link auf das neu erstellte Verzeichnis und die Installation der Binär-Distribution ist abgeschlossen.

```
> cd /usr/local
> mount /cdrom
```

```
> cp /cdrom/mysql-3.23.13a-alpha-pc-linux-gnu-i686.tar.gz
> # oder
> wget http://www.mysql.com/Downloads/MySQL-3.23/\
mysql-3.23.13a-alpha-pc-linux-gnu-i686.tar.gz
> tar zxvf mysql-3.23.13a-alpha-pc-linux-gnu-i686.tar.gz
> ln -s mysql-3.23.13a-alpha-pc-linux-gnu-i686 mysql
```

Sie könnten jetzt mit den nachfolgenden Befehlen eine Basis-konfiguration anlegen und den Datenbankserver starten.

```
> cd /usr/local/mysql
> scripts/mysql_install_db
> bin/safe_mysqld &
```

Rufen Sie die Programme genau *so* auf (wechseln Sie *nicht* z. B. ins Verzeichnis `scripts` um `mysql_install_db` von dort aus aufzurufen). `mysql_install_db` initialisiert bei Bedarf die MySQL-Benutzerverwaltung (das Skript läßt aber eine bereits eingerichtete Benutzerverwaltung unangetastet). `safe_mysqld &` startet den Datenbankserver und Sie können mit `/usr/local/bin/mysql -p root` den Standard-Komandozeilen-Client starten und sich im DBMS umschauen (geben Sie `help` ein). Sie sollten es bei dieser Konfiguration aber nicht belassen. Im übernächsten Abschnitt wird die Konfiguration des Datenbanksystems im Detail besprochen.

9.2.3 Installation der Source-Distribution

Auch die Installation der MySQL-Source Distribution bereitet in der Regel keine größeren Probleme. Sie bietet aber eine Reihe von Vorteilen. Wenn Sie sich mit der Installation von MySQL aus den Sourcen vertraut machen, sind Sie in der Lage, kurzfristig Sicherheitspatches einzuspielen. Sie können einen für deutsche Verhältnisse optimierten Zeichensatz einbinden. Sie können eine dynamisch gelinktes Programm erstellen (dies ist die Vor-

einstellung), so daß Sie auch benutzerdefinierte Funktionen (UDF, *user defined functions*) verwenden können. Sie können ferner das Installationslayout selbst bestimmen. Insgesamt bestehen mehr Einflußmöglichkeiten auf die Installation und es können bei Bedarf schnell und einfach Änderungen (auch im Sourcecode selbst) durchgeführt werden.

Nach dem Download des Distributionsarchives (bzw. nach seiner Kopie von der CD) z. B. in das Verzeichnis `/usr/local/src` muß zunächst wieder entpackt und ent-tart werden.

```
> cd /usr/local
> mount /cdrom
> cp /cdrom/MySQL/mysql-3_23_13a-alpha_tar.gz .
> # oder
> wget http://www.mysql.com/Downloads/MySQL-3.23/\
>   mysql-3.23.13a-alpha_tar.gz
> tar zxvf mysql-3_23_13a-alpha_tar.gz
> cd mysql-3.23.13a-alpha
```

Danach ist im Prinzip der Standard-Dreischritt (`./configure`, `make`, `make install`) durchzuführen. Es sollen aber nun einige Veränderungen von der Standard-Installation durchgeführt werden. Dies ist am einfachsten durch die Übergabe von entsprechenden Optionen an das Konfigurationsskript `configure` möglich. Mit `./configure --help` können Sie sich alle Optionen ausgeben lassen, die ausgewählt werden können. Im folgenden wird nur eine Auswahl besprochen.

Ausgewählte Optionen des MySQL-configure-Skripts	
Option [Voreinstellung]	Beschreibung
<code>--prefix=PREFIX</code> [<code>/usr/local</code>]	Hier kann ein vom Standard abweichende Installationsbasis eingestellt werden, wenn die Distributi-

	<p>on an einem anderen als dem voreingestellten Ort installiert werden soll.</p> <p>Beispiel: <code>--prefix=/usr/local/mysql</code></p>
<p><code>--bindir=DIR</code> [PREFIX/bin]</p>	<p>Wenn eine vom Standard abweichende Installationsbasis ausgewählt wurde (wie im Beispiel oben) kann hier dafür gesorgt werden, daß die Benutzer Binaries dennoch im Pfad landen.</p> <p>Beispiel: <code>--bindir=/usr/local/bin</code></p>
<p><code>--sbindir=DIR</code> [PREFIX/sbin] <code>--libexecdir=DIR</code> [PREFIX/libexec]</p>	<p>wie <code>--bindir</code>, nur daß hier der Pfad für System Administrator Binaries eingestellt wird (bzw für den MySQL-Server-Daemon <code>mysqld</code>, Dämonen wohnen gern in <code>sbin</code>).</p> <p>Beispiel: <code>--sbindir=/usr/local/sbin</code> <code>--libexecdir=/usr/local/sbin</code></p>
<p><code>--localstatedir=DIR</code> [PREFIX/var]</p>	<p>Hier kann eingestellt werden, wo die Datenbanken und die Log-Dateien abgelegt werden sollen. Bei SuSE-Linux werden solche Dateien z. B. in <code>/var/lib</code> abgelegt und wenn man es sich z. B. beim Backup wichtiger Daten nicht unnötig schwer machen will, empfiehlt es sich in der Regel, solche Standards einzuhalten.</p> <p>Beispiel: <code>--localstatedir=/var/lib/mysql</code></p>
<p><code>--mandir=DIR</code> [PREFIX/man] <code>--infodir=DIR</code> [PREFIX/info]</p>	<p>Die Pfade zur Standard-Dokumentation (die Manpages enthalten allerdings nur jeweils einen kurzen Überblick über die Commandline-Optionen der Programme).</p> <p>Beispiel: <code>--mandir=/usr/local/man</code> <code>--infodir=/usr/local/info</code></p>
<p><code>--with-unix-socket-path=absolute-file-name</code></p>	<p>Voreingestellt ist <code>, /tmp/mysqld.sock</code> was z. B. dann weniger erfreulich ist, wenn auf Ihrem System ein Cron-Job installiert ist, der regelmäßig das</p>

[/tmp/mysql.sock]	tmp-Verzeichnis von Überflüssigem befreit. Beispiel: --with-unix-socket-path= /var/lib/mysql/mysql.sock
--with-mysqld-user=username [root]	Der MySql-Daemon sollte <i>keinesfalls</i> als root laufen. Richten Sie einen Benutzer mysql ein, damit mysqld nicht unnötigerweise mit superuser-Rechten tätig wird. Beispiel: --with-mysqld-user=mysql
--with-low-memory [nicht gesetzt]	Gcc braucht zur Compilierung der Datei sql_yacc.cc riesige Mengen an Arbeitsspeicher. Sollten Sie über solche nicht verfügen, setzen Sie diese Option. (Falls Sie Probleme beim Compilieren bekommen, versuchen Sie diese Option auch dann, wenn Sie glauben, über ausreichend Speicher zu verfügen).
--with-charset=CHARSET [iso8859-1]	Standardmäßig wird der ISO8859-1 Zeichensatz bei der Darstellung und Sortierung verwendet. MySql beachtet deutsche Sortierregeln, wenn der Zeichensatz german1 angegeben wird (ein ISO8859-1-Zeichensatz mit deutschen Sortierregeln). Beispiel: --with-charset=german1

Aus verschiedenen unten näher beschriebenen Gründen verwenden wir folgende vom Standard abweichende Konfiguration:

```
> ./configure \
>   --prefix=/usr/local/mysql \
>   --localstatedir=/var/lib/mysql \
>   --mandir=/usr/local/man \
>   --infodir=/usr/local/info \
>   --with-unix-socket-path=/var/lib/mysql/mysql.sock \
```

```
> --with-mysqld-user=mysql \  
> --with-charset=german1
```

Als Installationsbasis verwenden wir das Verzeichnis `/usr/local/mysql`. Die gesamte Dokumentation von MySQL geht davon aus, das sich hier die Distribution befindet. Man tut sich bei Problemen leichter, wenn man gedanklich nicht jeweils abweichende Pfade verwenden muß. Für die Datenbanken und die Logdateien verwenden wir abweichend das Verzeichnis `/var/lib/mysql`, weil dies das Standardverzeichnis für Anwendungsdaten unter SuSE ist. In anderen Distributionen finden sich solche Dateien auch in `/usr/local/var/mysql` oder in `/var/opt/mysql`. Es ist eine gute Idee, sich in diesem Punkt an die Gepflogenheiten der Distribution zu halten. Sie müssen sonst möglicherweise bei der Planung eines Datensicherungskonzepts und der Implementierung einer Backup-Strategie zusätzlichen Aufwand treiben.

Die Manpages und die Dateien für GNU-Info möchten wir dagegen im hierfür vorgesehenen Standardverzeichnis plazieren (sonst können Sie später nicht z. B. `man mysql` angeben, sondern müssen so etwas wie `man -M /usr/local/man mysql` tippen, was sich auf Dauer als etwas lästig erweist). Das Socket-File möchten wir bitte auch nicht wie im Standard nach `/tmp` verfrachtet wissen, da unser Aufräumskript keine Gnade kennt und die Datei nächstens entfernt. Stattdessen soll `mysql.sock` dort zu finden sein, wo sich auch die Logfiles der Datenbank befinden. Wir möchten keinesfalls, daß der Datenbankdaemon mit `root`-Rechten läuft (weil dies ein erhebliches Sicherheitsloch darstellt) und konfigurieren daher für einen Benutzer `mysql`. Dieser sollte natürlich auf Ihrem System auch vorhanden sein. Legen Sie ihn ggf. neu an. (Der Benutzer `mysql` braucht kein Heimatverzeichnis, keine Shell und kein Kennwort). Wenn in Ihrer Datenbank deutsche Texte verwendet werden, ist es sinnvoll, den Zeichensatz `german-1` zu verwenden, weil dann Sortierungen entsprechend den deutschen Regeln durchgeführt werden.

Nachdem das `./configure` erfolgreich absolviert wurde, führen Sie noch das übliche:

```
> make
> make install
```

durch. Damit ist die Installation abgeschlossen. Sollten Sie Probleme bei der Kompilierung gehabt haben, versuchen Sie zunächst bei der Konfiguration die Option `--with-low-memory`. Auch wenn Sie glauben, über mehr als genug Speicher zu verfügen, versuchen Sie es ggf. mit dieser Option. Eine Reihe weiterer Probleme, die insbesondere durch die Umstellung von `libc` auf `glibc` hervorgerufen sind oder mit veralteten Libraries oder Programmen (z. B. `make`!) zusammenhängen, sollen hier nicht erörtert werden. Konsultieren Sie ggf. die wirklich hervorragende Dokumentation (die auch als HTML-Version auf der Begleit-CD enthalten ist). Obgleich auch wir uns aus Bequemlichkeit häufig an die erste Überlebensregel eines Unix-Systemadministrators halten "never touch a running system" stellt sich bei derart veralteten Systemen doch die Frage, ob hier nicht eine Radikalkur (sprich: Neuinstallation eines aktuellen Systems) insgesamt eher weniger als mehr Probleme bereitet.

Abschließend noch ein allgemeiner Hinweis zu Installations- bzw. Konfigurationsproblemen: Viele Schwierigkeiten entstehen dadurch, daß bei der Konfiguration von Software unter UNIX das Programm `configure` mehrfach aufgerufen wird (was an sich kein Problem ist), jeweils mit unterschiedlichen Optionen (weil beim Versuch zuvor etwas schief gegangen ist, z. B. Pfade nicht gefunden wurden und daher `configure` entsprechende Optionen mitgegeben werden sollen). Nur muß man wissen, das `configure` viele Informationen in der Datei `config.cache` abspeichert, die sie bei weiteren Aufrufen aus Geschwindigkeitsgründen verwendet, anstatt erneut erforderliche Prüfungen durchzuführen. Wenn man sich nicht ganz sicher ist, dann ist es daher eine gute Idee, diese Datei vor der erneuten Konfigurati-

on zu löschen. Auch das Aufräumen nach einem `make`-Lauf erspart häufig viele Probleme. Führen Sie also im Zweifel vor erneuten Konfigurations- und Compilierungsversuchen das Folgende aus:

```
> rm config.cache
> make clean
```

9.2.4 Basiskonfiguration

Bevor der Datenbankdaemon gestartet werden kann, muß, wenn MySQL zum ersten Mal installiert wird, das Benutzerrechte-System initialisiert werden. Dies erledigt das Skript `mysql_install_db`:

```
> cd /usr/local/mysql
> ./bin/mysql_install_db # oder
> ./skripts/mysql_install_db
```

Das Skript kann übrigens keinen Schaden anrichten. Falls Sie sich unsicher sind, ob die Rechteverwaltung bereits initialisiert ist, starten Sie es ruhig. Das Skript richtet einen Datenbankbenutzer `root` ein (der außer dieser Namensgleichheit nichts zu tun hat mit dem Unix-Superuser `root`). Der Datenbankadministrator `root` kann mit dem DBMS anstellen, was immer ihm beliebt und er kann (und muß zu diesem Zeitpunkt) sich *ohne Angabe eines Kennworts* einloggen. Wer merken uns das gewaltige Sicherheitsloch zur alsbaldigen Schließung vor! Außerdem erhält jede und jeder, die oder der sich ohne Angabe eines Benutzernamens und eines Kennworts beim Datenbanksystem anmeldet, Vollzugriff auf die Datenbank `test` sowie auf alle Datenbanken, die mit `test_` beginnen. Auch dies sollte man zugunsten einer gezielten, personen- und datenbankbezogenen Rechtevergabe alsbald ändern (vgl. Kapitel 9.2).

Lassen wir es indes zunächst dabei und starten den Datenbank-daemon:

```
> cd /usr/local/mysql
> /bin/safe_mysqld &
```

Wenn jetzt die Meldung ausgegeben wird:

```
Starting mysqld daemon with databases from /var/lib/mysql
```

dürften Sie es geschafft haben. Machen Sie gleich die Probe aufs Exempel indem Sie sich einige Informationen zum Befinden von MySQL und zu den Möglichkeiten, die Ihnen das Programm `mysqladmin` bietet, ausgeben lassen (statt `BINDIR` verwenden Sie das Verzeichnis, indem bei der Installation die Binaries abgelegt wurden, z. B. `/usr/local/mysql/bin`):

```
> BINDIR/mysqladmin -u root version
> BINDIR/mysqladmin -u root extended-status
> BINDIR/mysqladmin --help
```

Der Aufruf mit der Option `version` sollte etwa zu folgenden Ausgaben führen:

```
./mysqladmin Ver 8.3 Distrib 3.23.14-alpha,
  for pc-linux-gnu on i686
TCX Datakonsult AB, by Monty
```

```
Server version          3.23.14-alpha-log
Protocol version        10
Connection              Localhost via UNIX socket
UNIX socket             /var/lib/mysql/mysql.sock
Uptime:                 8 min 15 sec
```

```
Threads: 1 Questions: 42 Slow queries: 0 Opens: 8
Flush tables: 1 Open tables: 4
```

Der Aufruf mit der Option `extended-status` gibt eine Vielzahl weiterer Informationen aus (merken Sie sich diese Möglichkeit für später vor). Der `--help`-Aufruf zeigt u. a., daß man mit der Option `password` sein Kennwort ändern kann, daß mit der Option `shutdown` der Server heruntergefahren werden kann und daß noch eine ganze Reihe weiterer administrativer Aufgaben mit `mysqladmin` gelöst werden können.

Noch kann die Konfiguration aus Sicherheitsgründen allerdings nicht als befriedigend angesehen werden. Zunächst sollte auf jeden Fall der Zugang zum System durch den Datenbankadministrator `root` durch ein Kennwort geschützt werden. Dies kann z. B. so bewerkstelligt werden:

```
>BINDIR/mysqladmin -u root password topsecret
```

Mit diesem Aufruf setzen Sie für den Datenbankbenutzer `root` das Paßwort auf den Wert "topsecret". Der Aufruf `mysqladmin -u root version` sollte jetzt zu folgender Fehlermeldung führen:

```
> ./mysqladmin: connect to server at 'localhost' failed
error: 'Access denied for user: 'root@localhost' (Using
password: NO)'
```

Von nun an müssen Sie `mysqladmin` (wie alle anderen Hilfsprogramme einschließlich des Standard-Clients `mysql`) folgendermaßen aufrufen:

```
>BINDIR/mysqladmin -u root -p version
```

Der Parameter `-p` sorgt vor der weiteren Programmausführung für eine Abfrage des Paßworts. Wollen Sie Ihr Paßwort erneut ändern, geben Sie ein:

```
>BINDIR/mysqladmin -u root -p password strenggeheim
```

Sie werden nun zunächst nach Ihrem alten Paßwort gefragt (`topsecret`); nur wenn Sie dieses korrekt angeben, wird es auf `strenggeheim` geändert.

Als nächstes sollte dafür gesorgt werden, daß der Datenbank-daemon nicht wie voreingestellt mit root-Rechten läuft, sondern unter der Benutzerkennung `mysql` gestartet wird (schließlich hatten wir oben extra einen entsprechenden Benutzer eingerichtet). Zunächst sollte der Unix-Benutzer `mysql` Eigentümer des MySQL-Datenverzeichnis werden. Dazu müssen Sie den MySQL-Daemon zunächst herunterfahren, die Eigentumsverhältnisse ändern und anschließend den MySQL-Daemon wieder starten:

```
> BINDIR/mysqladmin -u root -p shutdown
> su -
> chown -R mysql /var/lib/mysql # Daten-Directory
> su mysql
> cd BINDIR/..
> BINDIR/safe_mysqld --user=mysql &
> exit
> ps aux | grep mysql
```

Der letzte Befehl prüft Ihre Bemühungen auf Erfolg und sollte zeigen, daß der Datenbankdaemon `mysqld` nunmehr dem Benutzer `mysql` und nicht mehr `root` gehört. Damit sind die größten Sicherheitsprobleme fürs Erste gelöst. Wenn Sie beabsichtigen, den Zugriff auf Ihre MySQL-Datenbank über das Internet (z. B. via PHP) zu gestatten, ist es hiermit allerdings nicht getan. In diesem Fall sollten Sie sich mit dem Thema Sicherheit eingehender beschäftigen. Ein gute Einführung zu diesem Thema bietet der Aufsatz "Webserver verstehen und tunen" von Kristian Köhntopp [koe99], der auch im Netz zur Verfügung steht.

Die weitere Konfiguration von MySQL können Sie in den Konfigurationsdateien `my.cnf` bzw. `.my.cnf` vornehmen. Eine Beispieldatei (`my-example.cnf`) finden Sie im Verzeichnis `support-files` der Source-Distribution (bzw. im Verzeichnis `/share/mysql` der Binär-Distribution). Wenn Sie eine sophisticated Konfiguration benötigen, gibt Ihnen MySQL mit seinem Drei-Ebenen-Kon-

zept der Konfigurationsdateien hierzu viele Möglichkeiten in die Hand. Zusammen mit der Empfindlichkeit, mit der MySQL gelegentlich auf Änderungen oder kleinere syntaktische Ungenauigkeiten in diesen Konfigurationsdateien reagiert, verschafft Ihnen das aber auch das Risiko, Sie tagelang beschäftigt zu halten. Bei Problemen mit sich sofort wieder verabschiedenden Servern und anderen Absonderlichkeiten denken Sie immer auch an diese Konfigurationsdateien. Ihre Verwendung durch Umzubenennung zu verhindern, ist im Problemfall oft das probate Mittel der Fehlerlokalisierung.

Globale Optionen können Sie in `/etc/my.cnf` ablegen, serverspezifische (das macht nur Sinn, wenn Sie mehrere Datenbankserver installieren wollen) in `DATADIR/my.cnf` (wobei `DATADIR` in unserem Beispiel `/var/lib/mysql` war), benutzerspezifische legen Sie in `~/my.cnf` ab. Diese drei Konfigurationsdateien werden in der Reihenfolge 1. globale Datei, 2. serverspezifische Datei, 3. benutzerspezifische Datei eingelesen. Es können also globale Optionen mit denen aus serverspezifischen Dateien und diese wiederum mit denen aus benutzerspezifischen Konfigurationsdateien überschrieben werden. In den Konfigurationsdateien können Sie jeweils Konfigurationsblöcke für die verschiedenen Programme der MySQL-Distribution anlegen. Dabei können alle Langoptionen (das sind die mit dem doppelten Bindestrich `--`) als Konfigurationsvariablen verwendet werden. Einen Überblick über diese Optionen verschafft Ihnen der Aufruf der Programme mit der Option `-help` (Für den MySQL-Daemon z. B. `mysqld --help`). Für Optimierungen bei hochfrequentierten oder sehr großen Datenbanken öffnet sich Ihnen hier ein reiches Betätigungsfeld. Im Rahmen dieses Kapitels kann hierauf nicht eingegangen werden (im Bedarfsfall finden sich Erläuterungen im MySQL-Handbuch im Abschnitt 11.2.3 Tuning Server Parameters).

Sinnvoll ist es in jedem Fall, mit Hilfe der globalen Konfigurationsdatei `/etc/my.cnf` (oder bei mehreren Servern mit der ser-

verspezifischen) das Logging zu aktivieren. Die Log-Dateien können auch in einem Backup-System eine wichtige Rolle spielen (Vgl. Abschnitt 9.3). Wichtig ist aber, daß Sie diese Dateien im Auge behalten (sie können sehr schnell sehr groß werden) und in ausreichend kurzen Abständen eine Logfile-Rotation durchführen. MySQL kann Ihnen seine Fehlermeldungen auch in Deutsch servieren. Wenn Ihnen das lieber ist als Englisch, können Sie auch dies hier aktivieren. Hier können Sie auch den Benutzer festlegen, mit dessen Rechten mysqld gestartet werden soll. Sie brauchen das dann nicht mehr beim Start des Servers per Hand eingeben. Eine globale Konfigurationsdatei könnte etwa so aussehen:

```
# Globale Konfiguratonnsdatei /etc/my.cnf
[client]
port                = 3306
socket              = /var/lib/mysql/mysql.sock

[mysqld]
port                = 3306
socket              = /var/lib/mysql/mysql.sock
skip-locking
set-variable        = key_buffer=16M
set-variable        = max_allowed_packet=1M
set-variable        = thread_stack=128K
log
language=german
user=mysql
```

Die `port` und `socket` Einträge sind nur wichtig, wenn Sie bei der Konfiguration hier Änderungen vorgenommen haben. Die `skip-locking` Direktive behebt ein Problem mit `fcntl()`-Aufrufen auf manchen Linux-Systemen. Die `set-variable` Direktiven entstammen der Beispiel-Konfigurationsdateien: `max_allowed_packet` muß mindestens so groß sein, wie Ihre größte in der Datenbank verwendete Spalte vom Typ `blob` oder

text. Wenn Sie den Wert für `key_buffer` heraufsetzen, können Indizes besser verwaltet werden. Auf einer Linux-Maschine mit 256Mbyte Arbeitsspeicher können Sie hier an Werte um 64M denken. Mit der Variablen `thread_stack` können Sie versuchen, die Performance zu verbessern, wenn Sie über reichlich Arbeitsspeicher verfügen. `log` sorgt für Logfiles (`DATADIR/server-name.log`).

Lassen Sie sich nicht die Einträge in der Beispiel-Konfigurationsdatei dazu verleiten, Kennworte in Ihrer `~/.my.cnf` abzuspeichern. Das erspart Ihnen zwar die Eingabe des Paßworts beim Aufruf des Clients, stellt aber ein Riesensicherheitsloch dar. Unproblematisch ist hingegen der Eintrag `user=irgendwer` im Client-Block. Er erspart die `-u` Option beim Aufruf der Clients.

9.2.5 Benutzung des Standard Client `mysql`

Der Standard MySQL Client `mysql` ist zwar ein völlig mausloses Kommandozeilen-Tool. Vielleicht läßt sich aber gerade deswegen schnell und effizient mit ihm arbeiten, wenn man einige wenige grundlegenden Funktionsweisen kennt. Da `mysql` die GNU-Readline-Bibliothek verwendet, werden Sie viele der Möglichkeiten von anderen Programmen (namentlich Emacs und Bash) her bereits kennen (und wenn nicht, sollten Sie sie jetzt unbedingt kennenlernen). Sie starten den Client mit dem Befehl:

```
> mysql -u root -p
```

Die Option `-p` sorgt dafür, daß Sie nach dem Paßwort gefragt werden. Wenn umfangreichere SQL-Statements in einer Datei abgelegt worden sind (z. B. Sicherheits-Dumps) und diese nun eingespielt werden sollen kann `mysql` auch als Filter verwendet werden, indem er wie folgt aufgerufen wird:

```
> mysql -u root -p database < dump.sql
```

Wenn in der Datei dump.sql zu Beginn das Statement use database; steht, kann man den Aufruf auf `mysql -u root -p < dump.sql` verkürzen. Wenn Sie die Ausgaben nicht auf den Bildschirm haben wollen, senden in eine Datei `ausgabe.datei`, dann geben Sie an:

```
> mysql -u root -p database < eingabe.sql >ausgabe.datei
```

Wenn interaktiv mit dem Client gearbeitet wird, stehen (unter anderem) die folgenden Tastenkürzel zur Verfügung (diese Tastenkürzel können übrigens auch in der Bash oder in Emacs verwendet werden):

Tastenkürzel und Kommandos im MySQL-Client mysql	
Pfeil hoch, Ctrl+P Pfeil runter, Ctrl+N	Ermöglicht es, auf zuvor eingegebenen Kommandos zuzugreifen (Scrollen durch die History)
Pfeil rechts Pfeil links	Cursor zurück /vor bewegen
Strg+A, Strg+E	Cursor an den Beginn / an das Ende der Zeile bewegen
Meta+F, Meta+B (Meta: häufig ESC)	Cursor ein Wort vor / zurück bewegen
Backspace, Entf	Zeichen rückwärts, vorwärts löschen
Strg+K	bis zum Ende der Zeile löschen
Strg+T	die beiden vorangegangenen Zeichen vertauschen
Meta+T (ESC+T)	die beiden vorangegangenen Worte vertauschen
Strg+L	säubert den Bildschirm
Tab	Expansion des Kommando oder Dateinamens (die Eingabe)

	<p>be von se[Tab] expandiert se zu select. Das funktioniert nicht nur mit SQL-Statements sondern auch mit Tabellen- und Spaltennamen.</p>
Strg+R	<p>Sucht nach bereits eingegebenen Kommandos: Drücken Sie am Beginn der Zeile Strg+R, und geben Sie dann die ersten Zeichen der gesuchten Befehlszeile ein. mysql zeigt daraufhin automatisch die erste passende Zeile an. Mehrmaliges Drücken von Strg+R wechselt zwischen verschiedenen passenden Möglichkeiten. Tab bricht die Suche ab und ermöglicht das Editieren der gefundenen Zeile.</p>
\c	<p>Eine Zeile, die mit Backslash c abgeschlossen wird, wird nicht beachtet (praktisch, wenn die Zeile wegen Fehler verworfen werden soll).</p>
\e	<p>Das eingegebene SQL-Statement wird in den Editor geladen, der in der Enviroment-Varibelen \$EDITOR genannt wird (das ist üblicherweis der VI). Praktisch, wenn Sie nach längerem Try and Error ein SQL-Statement optimiert haben und es nun in einer Datei sichern wollen.</p>
\. dateiname	<p>Mit dem Befehl \. können Sie ein SQL-Skript ausführen. Als Argument müssen Sie den Dateinamen angeben.</p>
\G	<p>Wenn Sie ein SQL-Statement mit diesem Befehl abschließen, wird das Ergebnis vertikal statt horizontal ausgegeben. Das ist besonders bei breiten Tabellen praktisch, bei denen man ansonsten schnell den Überblick verliert.</p>

Gewöhnlich geben Sie im mysql-Client einen oder mehrere SQL-Befehle ein, die anschließend an den Server geschickt werden. Nach Abarbeitung der Queries wird deren Ergebnis vom Server zurück- und vom Client auf den Bildschirm ausgegeben. Jedes SQL-Statement muß mit einem Semikolon abgeschlossen wer-

den. Solange der Client kein Semikolon sieht, geht er davon aus, das weitere Eingaben folgen. Beispiel:

```
mysql> use mysql;
mysql> SELECT
    -> host
    -> FROM
    -> user;
```

Wenn Sie eine Query doch nicht abschicken wollen, geben Sie einfach `\c` ein; mysql tut dann so, als hätte er nichts bemerkt:

```
mysql> SELECT
    -> host
    -> FR\c
mysql>
```

Für Verwirrung können nicht durch Anführungszeichen bzw. Hochkomma abgeschlossene Strings sorgen, weil sich das vermeintlich fertig gestellte SQL-Statement weder durch ein Semikolon abschicken, noch die Eingabe durch `\c` abbrechen läßt:

```
mysql> SELECT user FROM user WHERE host="
    "> \c
    "> ;
    "> "\c
mysql>
```

Nach dem Ende der ersten Zeile erwartet der Client einen String, daher verliert sowohl `\c` als auch `;` seine Sonderbedeutung. Erst nachdem `mysql` das Ende des Strings durch abschließende Anführungszeichen bedeutet wird, erkennt der Client wieder die Sonderbedeutung von `\c` und bricht die Verarbeitung ab. Das funktioniert genauso mit einfachen Hochkomma, der Client gibt dann ein aufforderndes `'>` statt des `">` aus.

Abschließend sei noch darauf hingewiesen, daß Sie den MySQL-Client durchaus auch benutzen können, um Datenbanken zu

administrieren, die auf Servern irgendwo im Internet gehostet werden (sofern Sie eine Zugangsberechtigung besitzen und die entsprechenden Ports offen sind). Rufen Sie dazu einfach `mysql` mit der Option `-h` auf (z. B. `mysql -h fremder.host.de -u user-name -p`) auf.

9.2.6 Hilfsprogramme der MySQL-Distribution

Im folgenden wird auf einige Hilfsprogramme und Skripts der MySQL-Distribution aufmerksam gemacht. Die Programme werden nicht vollständig besprochen, sondern nur ihre allgemeine Funktion genannt. Genaueres können Sie dem MySQL-Handbuch entnehmen. Fast alle Tools geben außerdem mit der Option `--help` einen Überblick über ihre Funktion und die verfügbaren Commandline-Switches aus. Gelegentlich geben wir Hinweise auf besondere (oder versteckte) Features.

mysql als Kommandozeilen-Filter

Der Standard-Client `mysql` kann, wie oben bereits gezeigt, auch als Commandline-Filter verwendet werden. In diesem Fall erhält `mysql` seine Eingaben von Standard-In und schreibt Ausgaben auf Standard-Out. Diese Ausgaben können Sie in üblicher UNIX-Manier z. B. in Dateien umleiten oder in andere Programme pipen und natürlich auch in Shellskripts oder in Skripts anderer Interpretersprachen wie Perl, PHP oder Python verwenden. Bei der Verwendung von `mysql` als Filter sind mitunter die folgenden Commandline-Switches von nützlich:

- B**, **--batch**: Gibt die Ergebnisse einer Query als Tab Delimited Data aus. Jede Spalte wird durch ein Tabulatorzeichen von der nächsten getrennt, jede Zeile von der nächsten durch ein Newline. Dies ist ein Format, mit dem viele Anwendungsprogramme (z. B. Tabellenkalkulationen) problemlos umgehen können.
- E**, **--vertical**: Gibt die Ergebnisse vertikal aus. Es wird keine Tabelle erzeugt, sondern es wird eine Ausgabe in folgender Form erzeugt:

```

...
Alter_priv: Y
***** 9. row *****
Host: %
Db: mysql
User: niemand
Select_priv: Y
Insert_priv: N
Update_priv: N
Delete_priv: N
Create_priv: N
Drop_priv: N
Grant_priv: N
References_priv: N
Index_priv: N
Alter_priv: N
***** 10. row *****
Host: localhost
...

```

- f, **--force**: Normalerweise bricht mysql die Verarbeitung eines Skripts mit einer Fehlermeldung ab, wenn der Client auf ein ungültiges Statement trifft. Die Option **--force** sorgt dafür, daß nach einem Fehler die Abarbeitung des Skripts fortgesetzt wird.
- H, **--html**: Mit dieser Option produziert der Client seine Ausgabe im HTML-Format. Es werden Standard HTML Tabellen (<table></table>) erzeugt. Als schnelle Lösung für den internen Gebrauch oder zum Einbetten in eine HTML-Seite durchaus zu gebrauchen.
- L, **--skip-column-names**: Unterdrückt die Ausgabe der Spaltennamen.
- p, **--password**: Diese Option kann auch mit dem Kennwort als Argument verwendet werden. Von dieser Möglichkeit sollte man aus Sicherheitsgründen nur dann Gebrauch machen, wenn dies unumgänglich ist. Bei Verwendung der kurzen Form (-p) darf zwischen -p und dem Paßwort *kein* Leerzeichen stehen.
- t, **--table**: Die Ausgabe erfolgt wie bei der interaktiven Ausgabe im ASCII-Tabellenformat:

```

+-----+-----+
| host      | user      |
+-----+-----+
| %         | niemand   |
| localhost | niemand   |
+-----+-----+

```

mysqladmin

Auf einige Funktionen von mysqladmin wurde bei der Diskussion der Serverkonfiguration bereits eingegangen. Hier ein kurzer Überblick:

mysqladmin - Aufruf: <code>mysqladmin -u root -p option argument</code>	
Optionen	Beschreibung
create dbname drop dbname	Erstellt eine Neue Datenbank dbname bzw. löscht eine bestehende samt aller Tabellen.
version variables status extended-status	Gibt mehr oder weniger umfängliche Informationen über die Befindlichkeit und den Zustand des Servers aus.
processlist kill id,id,..	Zeigt Informationen (Id, user, host, ...) zu allen aktiven Prozessen auf dem Datenbankserver. Mit kill können einzelne Prozesse abgeschossen werden.
ping shutdown	Ping prüft, ob der Datenbankserver wohlauf ist. Mit Shutdown wird der Server heruntergefahren.
password	Ändert ein bestehendes Paßwort. Als Argument wird das neue Kennwort angegeben.

reload flush-privileges	Liest die Benutzerauthentikations-Tabellen (grant tables) neu ein. Notwendig, wenn diese Tabellen "händisch" geändert wurden.
flush-hosts flush-tables	Löscht interne Caches. flush-hosts ist z. B. notwendig, wenn einige Ihrer Hosts ihre IP-Nummer geändert haben.
flush-logs	Initialisiert die Logfiles neu (vgl. Abschnitt 9.3)
refresh	Führt flush-tables und flush-logs aus.

mysqldump

Mit diesem Utility können Sie eine Datenbank oder eine ganze Sammlung von Datenbanken "dumpen" (abladen, abspeichern). Der Dump besteht aus SQL-Statements, die Tabellen anlegen und füllen (CREATE TABLE, INSERT). Diese Dumps können als Sicherheitskopien dienen und eignen sich auch, um Datenbanken über Betriebssystemsgrenzen hinweg auszutauschen. Nach einigen Anpassungen gelingt damit in der Regel auch der Austausch von Datenbanken zwischen verschiedenen DBM-Systemen (z. B. zwischen MySQL und Oracle oder Informix). Eine umfassende Referenz bietet das Handbuch (bzw. die Option --help), hier nur einige Hinweise zu einzelnen Comandline-Switches.

Aufruf: `mysqldump -u root -p --opt dbname > backup.sql`

-q, --quick: Wenn Sie diese Option *nicht* angeben, lädt MySQL zunächst die gesamte Ausgabe in den Arbeitsspeicher, bevor gedumpt wird. Wenn Sie eine größere Datenbank dumpen wollen oder über wenig Arbeitsspeicher verfügen, bekommen Sie damit ein Problem.

--opt: Beinhaltet `--quick` und erstellt das schnellst mögliche Dump. Dieses eignet sich dann aber nur für einen MySQL-Server (weil Nicht-Standard-Features von MySQL verwendet werden).

-d, --no-data: Gibt nur die Tabellenstruktur einer Datenbank, nicht aber deren Inhalt aus.

--all-databases: Erstellt einen Dump mit allen Datenbanken.

Wenn Sie darüber nachdenken, Ihre Datenbanken auf einem anderen Host zu kopieren (z. B. aus Sicherheitsgründen), dann versuchen Sie doch einmal den folgenden Aufruf:

```
> mysqldump -opt -all-databases | \  
>   mysql -host=www.ihr-mirror.de -compress
```

mysqlimport

Mit `mysqlimport` können Sie strukturierte Daten aus anderen Anwendungen in MySQL importieren. `mysqlimport` bedient sich intern des `LOAD DATA INFILE SQL`-Statements und stellt dessen komplette Funktionalität für die Kommandozeile zur Verfügung. `LOAD DATA INFILE` stellt ein flexibles Instrument dar, auch ungewöhnlich formatierte Daten in MySQL zu importieren. Nicht möglich ist derzeit allerdings das Einlesen von Daten, bei denen die Zugehörigkeit zu einer Spalte durch die Position in einer Zeile definiert wird, wie dies bei der Datenerfassung für Statistikprogramme üblich ist (z. B. Identnummer: Zeichen 1-6, Vorname: Zeichen 7-30, Nachname: Zeichen 31-55 usw.). In einem solchen Fall müssen Sie z. B. mit einem entsprechenden Perl-Skript vorarbeiten.

Weitere Hilfsprogramme

myisamchk / (isamchk): Programm, mit dem MySQL-Tabellen geprüft, optimiert und repariert werden können. `isamchk` wird für Tabellen im älteren Tabellentyp (vor MySQL 3.23.6) verwendet. Zwischen altem und neuem Typ kann problemlos gewechselt werden. Genaueres finden Sie im Abschnitt 9.3.

mysqlshow: Ein Commandline-Interface zum SQL-Befehl SHOW. Mit diesem Befehl können Sie sich Informationen zu Datenbanken, Tabellen, Spalten und Indizes ausgeben lassen.

mysqlbug: Das MySQL-Fehler-Dokumentations-Skript. Wenn Sie bei den Entwicklern von einem Programmfehler melden wollen, müssen Sie dieses Skript verwenden. Das Programm fragt eine Reihe von Systemparametern und die MySQL-Konfiguration ab und schreibt diese Informationen in eine Datei, die das Programm auch gleich in Ihrem Editor (z. B. Emacs) öffnet. Sie sollten dieses Skript auch verwenden, wenn Sie eine Frage in die MySQL-Mailingliste posten. Die Informationen erleichtern häufig die Fehlerdiagnose.

mysql2mysql: Wenn Sie von mSQL auf MySQL umsteigen wollen, schauen Sie sich dieses Tool an. Es kann nicht alles konvertieren, wird Ihnen aber die Arbeit stark erleichtern.

mysqlaccess: Prüft das Benutzer-Authentikations-System von MySQL. Näheres finden Sie im Abschnitt 9.2.

mysql_install_db: Intitalisiert das Benutzer-Authentikations-System von MySQL. Wird nur bei der Erstinstallation von MySQL benötigt (vgl. Kapitel XX).

safe_mysqlld: Dieses Skript startet den MySQL-Server und führt zuvor einige Prüfungen durch (vgl. Kapitel XX).

9.2.7 Webmin

Webmin ist ein Web-basiertes Interface für Systemadministratoren unter Unix. Mit Hilfe eines Standard Web-Browsers können User Accounts gewartet, der Web-Server Apache konfiguriert, das Domain Name Service (DNS) gepflegt und - das ist der Grund, warum hier auf dieses Tool eingegangen wird - der Datenbankserver MySQL administriert werden. Webmin kann noch viel mehr (z. B. Samba und Sendmail konfigurieren, RAID erstellen, Cron Jobs einrichten); davon soll hier aber nicht die Rede sein. Schauen Sie sich das Tool nach der Installation einfach mal genauer an! Webmin besteht aus einem einfachen Webserver und einer Reihe von CGI Programmen, die direkt Systemdateien wie /etc/inetd.conf oder /etc/passwd manipulieren. Der Webserver und alle CGI Programme sind in Perl geschrieben.

Wenn Sie Webmin verwenden wollen, müssen Sie erstens eine Linux-Distribution verwenden, die von Webmin unterstützt wird (es werden sehr viele Distributionen unterstützt: Caldera, Redhat, Slackware, Debian, SuSE, Corel, TurboLinux, Cobalt, Mandrake, Delix, MxLinux sowie eine Reihe anderer Betriebssysteme einschließlich Sun Solaris, weiteres vgl. <http://www.webmin.com/webmin/support.html>). Zweitens muß Perl in der Version 5 auf Ihrem Rechner installiert sein. Perl Vers. 5 ist unter Linux Standard und wird von allen uns bekannten Distributionen unterstützt und daher hier nicht weiter besprochen. (Auch wenn Sie nicht wissen, ob Perl auf Ihrem Rechner installiert ist, wird es wahrscheinlich vorhanden sein: probieren Sie mal den Befehl `perl -v`). Webmin ist Open Source und kann - kommerziell wie nicht-kommerziell - frei verwendet werden. Webmin ist ferner erweiterbar: Wer will (und kann), kann eigene Module für Webmin schreiben. Eine Auswahl finden Sie unter <http://www.webmin.de/webmin/third.html>.

Webmin kann daneben auch SSL (*Secure Socket Layer*) nutzen, um sichere Verbindungen zwischen Ihrem Webbrowser und dem Webmin Server herzustellen. Das funktioniert allerdings nur, wenn die Perl `Net::SSLeay` Module installiert sind. Und diese wiederum benötigen zwingend die OpenSSL C Libraries. Aus Sicherheitsgründen ist sehr dringend die Nutzung dieses Features anzuraten, sofern Sie nicht ausschließlich lokal arbeiten, ohne je Verbindung ins Internet aufzunehmen.

Fangen wir also sogleich mit der Installation von OpenSSL an (OpenSSL ist nicht Standard, die Installation soll daher kurz dargestellt werden). OpenSSL wird vom „OpenSSL Core and Development“ Team entwickelt (einem Zusammenschluß von Freiwilligen aus England, Deutschland und Schweden) dessen Ziel es ist, eine leistungsfähige Implementation des *Secure Sockets Layer* (SSL v2/v3) und des *Transport Layer Security* (TLS v1) Protokolls sowie ein allgemein nutzbare Cryptographie-Bibliothek zu erstellen. OpenSSL ist Open Source Software, die für

kommerzielle wie nicht-kommerzielle Zwecke frei verwendet werden kann. Sie erhalten OpenSSL von der OpenSSL-Website (<http://www.openssl.org/>). Die Version 0.9.5, deren Installation im folgenden besprochen wird, finden Sie natürlich auch auf der Buch-CD.

Nach dem Download bzw. der Kopie von CD in das Installationsverzeichnis muß die Software zunächst wie üblich entpackt werden:

```
# ins Installationsverzeichnis wechseln
> cd /usr/local/src
> # entweder mit cp von CD oder mit wget aus dem Internet
> # die Source holen
> mount /cdrom
> cp /cdrom/openssl/openssl-0_9_5_tar.gz . # oder
> wget http://www.openssl.org/source/openssl-0.9.5.tar.gz
> # archiv auspacken
> gunzip openssl-0.9.5.tar.gz
> tar xvf openssl-0.9.5.tar
> cd openssl-0.9.5
```

Es folgt der übliche Dreiklang: Konfiguration für das Betriebssystem mit configure, Compilierung der Source (Makefile gesteuert), Installation der Binärdateien:

```
# Configure starten
> ./config
```

Wenn Sie erfolgreich waren, sollte am Ende einer ellenlanger Ausgaben eine Meldung wie die folgende erscheinen:

```
Configured for linux-elf.
```

Als nächstes werden die Binaries compiliert:

```
> make
# war alles erfolgreich, rufen sie
```

```
# make test auf (am Besten wie folgt, dann können sie nach
# Ende des Skripts in der Datei maketest.txt nachlesen,
# was da gerade am Bildschirm vorbeigerauscht ist ;-)
> make test | tee maketest.txt
```

Wenn das Studium der `maketest.txt` Datei ergab, daß diese neben Unverständlichem auch jede Menge `ok` enthielt und keine `fatal error` Meldung oder ähnliches, ist es Zeit für die Installation. Installations-Directory ist (aus historischen Gründen) standardmäßig `/usr/local/ssl`.

```
# Installation als root durchführen
> su -
  <rootpassword>
> make install
> exit
```

Und fertig! Wenn Sie jetzt gerade nichts Besseres zu tun haben, können Sie die Manpages zu OpenSSL studieren

```
> man -M /usr/local/ssl/man openssl
```

und sich darüber informieren, was Sie sonst noch so alles mit OpenSSL anstellen können. Danach geht es weiter mit der Installation des Perl Net::SSLeay Moduls. Es beginnt wieder der übliche Ablauf: Source kopieren, auspacken, ins Source-Verzeichnis wechseln, compilieren und testen und – sofern alles gut ging – als root installieren. Das Compilieren und Testen wird vom Befehl `Makefile.PL -t` in einem Aufwasch erledigt.

```
> cd /usr/local/src
> # entweder mit cp von CD oder mit wget aus dem Internet
> # die Source holen
> cp /cdrom/Net_SSLeay/Net_SSLeay.pm-1.05.tar.gz . # oder
> wget http://www.cpan.org/modules/by-module/Net/\
>   Net_SSLeay.pm-1.05.tar.gz
> gunzip <Net_SSLeay.pm-1.05.tar.gz | tar xvf -
```

```
> cd Net_SSLeay.pm-1.05
> ./Makefile.PL -t
> su -
  <rootpassword>
> make install
  ...
> exit
```

Lassen Sie sich ggf. nicht von der folgenden Fehlermeldung irritieren:

```
Your vendor has not defined SSLeay macro xx
at test.pl line xx
```

SSLeay ist ein mit OpenSSL geliefertes Makro, das von Net::SSLeay (trotz des Namens) *nicht* (mehr) unterstützt wird. Eine Reihe von Tests werden daher mit der genannten Fehlermeldung abgebrochen. Tun wir so, als hätten wir nichts bemerkt. Sollten sich andere Fehlermeldungen einstellen, bleibt Ihnen nichts, als sich durch das README zu kämpfen und/oder die Homepage von Net::SSLeay aufzusuchen (http://www.bacus.pt/Net_SSLeay/index.html).

Nach all diesen Vorarbeiten (die nicht zwingend notwendig sind für den Betrieb von Webmin aber aus Sicherheitsgründen dringend anzuraten), beginnen wir die Installation von Webmin. Da es sich bei Webmin ausschließlich um Perlskripts handelt, braucht nichts compiliert zu werden. Die Installation verläuft daher insofern etwas anders, als wir das Paket direkt ins Installationsverzeichnis kopieren (gewöhnlich `/usr/local`), dort (als user root!) entpacken und dann bereits durch Aufruf des Shellskripts `setup.sh` die Konfiguration beginnen.


```
> cd /usr/local
> su [RETURN] <rootpassword>
> # entweder mit cp von CD oder mit wget aus dem Internet
> # die Source holen
```

```
> cp /cdrom/webmin/webmin-0.78.tar.gz . # oder
> wget http://www.webmin.de/webmin/download/\
> webmin-0.78.tar.gz
> gunzip < webmin-0.78.tar.gz | tar xvf -
> cd webmin-0.78
> ./setup.sh
```

Das Skript befragt Sie nun ausgiebig und es empfiehlt sich, wahrheitsgemäß zu antworten. Webmin möchte zunächst wissen, wohin es seine Konfigurationsdateien speichern soll. Sinnvollerweise wählen Sie hier nicht das Installationsverzeichnis von Webmin sondern das Verzeichnis, indem in Ihrer Distribution gewöhnlich Konfigurationsdateien gelagert werden (in der Regel `/etc/webmin`). Wenn Sie zukünftig ein Update von Webmin erwägen sollten, bleibt nur so Ihre vorherige Konfiguration erhalten. Als „Webmin log directory“ (dies ist die nächste Frage) wählen Sie unter SuSE Linux üblicherweise `/var/webmin`. In dieses Verzeichnis speichert Webmin seine pid und ggf. Logdateien des Webservers. Weiter möchte Webmin über den Fundort Ihres Perl-Binary informiert werden: Das Kommando `whereis perl` gibt hierüber Auskunft (in der Regel findet sich Perl in `/usr/bin/perl`). Nachdem Sie Ihr Betriebssystem angegeben sowie einen TCP-Port, auf dem Ihr Webminserver lauschen soll (Vorschlag: 10000), müssen Sie sich ein Loginnamen und ein Paßwort ausdenken, mit dem Sie Zugriff auf den Webmin Webserver haben wollen. Sie sollten dieses Paßwort genauso sorgfältig auswählen wie Ihr Rootpaßwort. Wer Zugang zum Webminserver hat, kann auf Ihrem Server treiben, was immer ihm paßt (einschließlich `rm -r / !`).

Aus diesem Grund sollten Sie den Zugriff auch begrenzen auf Anfragen, die von vertrauenswürdigen Hosts kommen. Dies können Sie nach der Installation auf den Konfigurationsseiten von Webmin erledigen. Schließlich will das Installationskript noch den Namen des Servers wissen, auf dem Sie gerade Webmin installieren. Außerdem werden Sie gefragt, ob Sie SSL akti-

vieren wollen. Sie wollen (sonst hätten Sie sich ja die Mühe mit der Installation von OpenSSL und Net::SSLeay sparen können). Wenn Sie nach der Installation Schwierigkeiten haben, auf den Webminserver zuzugreifen, prüfen Sie zunächst, ob Sie den URL korrekt angegeben haben (es muß heißen: <https://ihr-server.de:10000/>; Sie haben das kleine „s“ bei https wahrgenommen?). Klappt es immer noch nicht, editieren Sie die Datei `/etc/webmin/miniserv.conf` und ändern den Eintrag `ssl=1` (9. Zeile) in `ssl=0`. Damit schalten Sie die zusätzliche Sicherheit durch die SSL-Verschlüsselung wieder ab. Anschließend gehen Sie zu Ihrem Systemadministrator und beschweren sich über das restriktive Firewall oder/und den unmöglichen Proxy (Wenn Sie selbst der Systemadministrator sind wissen Sie ja, was zu tun ist). Zu guter Letzt sollten Sie sich noch dafür entscheiden, daß der Webminserver beim Bootvorgang automatisch gestartet wird (sonst müssen Sie das nach jedem Bootvorgang von Hand machen). Auch das erledigt das Installationskript für Sie. Zum Schluß sagt Ihnen das Skript, unter welchem URL Sie den Webminserver aufrufen können (z. B. <https://ihr-server.de:10000/>).



mysql001.tif/psd

Abb. 9.1: Webmin, das Tool für den Systemadministrator.

Der Lohn Ihrer Installationsmühen wird Ihnen nach dem Aufruf von <https://ihr-server.de:10000/> noch eine kleine Weile vorenthalten: Zunächst weist Sie Ihr Browser daraufhin, daß ihm das vom Server vorgelegte Zertifikat unbekannt sei. Dies liegt daran, daß Webmin SSL nur für die Verschlüsselung verwendet und nicht zur Authentizierung. Sagen Sie „ja“, wenn Sie gefragt werden, ob Sie das Zertifikat akzeptieren wollen. Haben Sie sich gegen SSL entschieden, rufen Sie den URL <http://ihr-server.de:10000/>, http ohne „s“, auf und haben keine weiteren Scheereien (aber auch keine erhöhte Sicherheit). Nun endlich präsentiert sich Ihnen der Webmin-Startbildschirm (vgl. Abb. 9.1).

Am geschlossenen Vorhängeschloß (2) erkennen Sie, daß Ihre Bemühungen um eine abhörsichere Verbindung zum Server erfolgreich waren. Als nächstes sollten Sie (und zwar jetzt gleich und sofort, wegen der Sicherheit) zur `WEBMIN CONFIGURATION` wechseln (3).

`mysql002.tif/psd`

Abb. 9.2: Konfiguration von Webmin.

Zuallererst sollten Sie den Menüpunkt `IP ACCESS CONTROL` aufrufen und auf dem nachfolgenden Bildschirm die Option `ONLY ALLOW FROM LISTED ADDRESSES` auswählen. Natürlich müssen Sie dann auch (mindestens) eine IP-Adresse angeben (sonst werden Sie selbst nicht mehr auf Webmin zugreifen können). Falls Sie Ihre IP-Adresse nicht kennen, können Sie diese mit dem Befehl `ifconfig` herausfinden. Sie können natürlich auch mehrere Adressen angeben (falls Sie von verschiedenen Rechnern aus zugreifen oder anderen ebenfalls den Zugriff ermöglichen wollen). Es ist auch möglich, eine Netzwerkmaske anzugeben und so z.B. Ihr lokales Firmennetzwerk generell für vertrauenswürdig zu erklären (z. B. erlaubt der Eintrag `192.124.237.0` allen 254 möglichen Adressen des entsprechenden Klasse-C-Netzwerks den Zugriff).

Das bedeutet freilich nicht, daß jede und jeder Zugriff auf Webmin hat, die oder der von einem vertrauenswürdigen Host kommt. Dies ist lediglich die Mindestvoraussetzung. Zusätzlich muß dem Begehrlichen auch der Username und das Passwort bekannt sein, das Sie bei der Installation angegeben haben. Sie können übrigens weitere Nutzer einrichten (über die Option `WEBMIN USERS`), deren Zugriffsmöglichkeiten Sie auch beschränken können (z. B. könnten Sie einen Account `DBAdmin` einrichten, dessen Nutzern Sie lediglich den Zugriff auf das MySQL-Modul erlauben). Wenn Sie schon mal beim Konfigurieren sind, können Sie auch gleich die Sprache auf `GERMAN (DE)` einstellen (nach Aufruf der Option `LANGUAGE (3)`).

mysql003.tif/psd

Abb. 9.3 Webmin Menü: System, Hardware, Others.

Abb. 9.3 gibt einen Überblick über die sonstigen Möglichkeiten von Webmin. Wer manchmal heimlich eine Maus benutzt, wird sich darüber freuen, viele administrative Aufgaben durchführen zu können, ohne jede einzelne der Linuxkonfigurationsdateien auswendig beim Namen nennen zu können. Außerdem ermöglicht Webmin die Fernwartung und -kontrolle auf einfache und elegante Weise. Ob Sie schnell einmal die Logfiles überprüfen, die Disk-Quota eines nörgelnden Users heraufsetzen, einen neuen Benutzer einrichten oder gar ein RAID-System einrichten und konfigurieren wollen: Webmin steht Ihnen zur Seite.

mysql004.tif/psd

Abb. 9.4 Webmin Menü: Servers.

Am interessantesten ist aber zweifellos das Menü `SERVERS`. Ob es um die (nicht triviale) Konfiguration des Webservers Apache geht, mit der sich jede und jeder befassen muß, die oder der eine Datenbankanbindung ins Web bauen will oder muß (wir gehen hierauf noch bei der Diskussion von PHP im Kapitel 13 ein), ob es die Verwaltung eines Domain Name Systems ist oder der Aufbau und die Verwaltung eines FTP-Servers (WuFTP) oder die (auch nicht triviale) Einrichtung und Konfiguration der Kommunikation mit der Windows-Welt (Administration von Samba via `WINDOWS DATEIFREIGABE`): für alle diese Aufgaben bietet Webmin seine Unterstützung an. Sogar die auch unter alten Unix-Hasen gefürchtete Sendmail-Konfiguration wird mit weiteren vierzehn Untermenüs (auf die Sie nach Klick auf `SENDMAIL KONFIGURIEREN` zugreifen können) stark vereinfacht. Für uns ist natürlich die Administration des Datenbankmanagementsystems MySQL von besonderem Interesse. Sie erreichen das im Abb. 9.5 gezeigte Menü durch Klick auf `MYSQL DATENBANK SERVER`.

Im oberen Abschnitt des Bildschirms werden alle von Ihnen angelegten Datenbanken aufgeführt. Auch wenn Sie noch keine Datenbank erstellt haben werden Sie hier mindestens eine, wahrscheinlich zwei Datenbanken vorfinden: `mysql` und `test`. Die Datenbank `mysql` ist eine vom DBMS MySQL zu administrativen Zwecken angelegte Verwaltungsdatenbank (mit der man sich in jedem Fall noch intensiver beschäftigen wird müssen, die ultimative Aufklärung zur Funktionsweise dieser Datenbank erhalten Sie im Kap. 9.2 Benutzerverwaltung). Die Datenbank `test` wird ebenfalls bei der Installation angelegt. Sie können sie bedenkenlos löschen oder in Ihr Tabellen anlegen. Schreiten Sie ruhig einmal zur Tat. Der Klick auf `test` öffnet einen Bildschirm auf dem Sie beherzt den Menüpunkt `ERSTELLE EINE NEUE RELATION` auswählen. In der sich öffnenden Dialogbox geben Sie einen Tabellennamen an (unter „Relationsname“, z. B. `adressen`) und legen vier Felder an (z. B. `nachname`, `vorname`, `telefon`, `email`). Als Datentyp wählen Sie für alle vier Felder `varchar` und geben für jedes Feld eine maximale Anzahl Zeichen an. Noch ein Klick auf Erstellen schon ist eine neue Tabelle (Relation) in MySQL erstellt. Wenn Sie die Tabelle anklicken, gelangen Sie zu einem Bildschirm, indem Sie diese wieder löschen können oder auch weitere Felder hinzufügen oder sich die bisher eingegebenen Daten anschauen können (da Sie noch keine Daten eingegeben haben, werden Sie allerdings nichts sehen). Unter dem Menüpunkt `SQL AUSFÜHREN` können Sie beliebige SQL-Befehle an die Datenbank schicken.

Das MySQL-Modul von Webmin eignet sich besonders zur Administration des Rechtesystems von MySQL. Allerdings müssen Sie wissen, was Sie tun. Dieses Wissen vermittelt Ihnen Webmin nicht. Deshalb übernehmen wir das in Kapitel 9.2. Mit Webmin können Sie sich darüber hinaus schnell einen Überblick

über die von Ihnen angelegten Tabellen verschaffen. Für die Detail-Administration einzelner Datenbanken empfehlen wir Ihnen hingegen phpMyAdmin. Dieses hervorragende Tool wird im nächsten Kapitel besprochen.

9.2.8 Web-Browser als Client: phpMyAdmin

Der MySQL-Client phpMyAdmin ist – wie Webmin – eine Anwendung für Ihren Webbrowser. Sie installieren phpMyAdmin auf dem Server, auf dem Ihre MySQL-Datenbank gehostet ist, und sind dann in der Lage, mit einem Web-Browser von einem beliebigen Ort aus auf Ihre Datenbanken zuzugreifen. phpMyAdmin ist ein wirklich fortgeschrittenes und ausgereiftes Tool (Vielen Dank an den Autor Tobias Ratschiller!). Es gibt kaum ein administratives Problem, daß sich nicht mit ihm lösen ließe. Die Installation lohnt sich daher auch dann, wenn Sie vollen Zugriff auf Ihr Linuxsystem haben und der „Umweg“ über das Browserinterface eigentlich nicht nötig ist. Momentan können Sie folgende administrativen Aufgaben mit phpMyAdmin durchführen:

- Erstellen und Löschen von Datenbanken
- Tabellen erstellen, kopieren, löschen und ändern
- Felder löschen, hinzufügen und ändern
- Ausführung aller (in MySQL implementierten) SQL-Befehle. Eine Batch-Verarbeitung von mehreren SQL-Statements ist ebenfalls möglich.
- Administration von Schlüsseln
- Einladen von strukturierten ASCII-Dateien (CSV, Tab-delimited usw.) in Tabellen

- Erstellen von SQL-Dumps (das sind SQL-Befehle wie CREATE TABLE und INSERT, die eine Tabelle oder eine gesamte Datenbank samt ihren Inhalts restaurieren)
- Möglichkeit des Uploads von SQL-Dumps (!)
- Datenexport im CSV-Format

phpMyAdmin ist ein Open Source Projekt und unterliegt der GNU General Public License. phpMyAdmin ist vollständig in PHP geschrieben (falls Sie nicht wissen was PHP ist: PHP ist eine Programmiersprache, die entwickelt wurde, um interaktive Webangebote zu ermöglichen und häufig als Middleware zwischen Webserver und Datenbank eingesetzt wird, vergleiche hierzu ausführlich Kapitel 13). Sie können daher mit phpMyAdmin nichts anfangen, wenn auf Ihrem System PHP nicht installiert ist. Außerdem muß PHP mit MySQL-Support compiliert worden sein. Möglicherweise ist dies aber bereits der Fall: Bei SuSE 6.1 und höher wird PHP standardmäßig installiert, wenn Sie den Apache Webserver installieren. Sie können dies testen, indem Sie in das Document Root Verzeichnis des Apache Webservers eine Datei `php_info.php3` mit folgendem Inhalt einstellen (heben Sie sich diese Datei auf, Sie werden sie immer wieder einmal benötigen, wenn Sie wissen möchten, wie Ihr Webserver bzw. Ihre PHP-Installation konfiguriert ist):

```
<html>
<head><title>PHP-Info</title></head>
<body>
<?php echo phpinfo() ?>
</body>
</html>
```

Rufen Sie nun in Ihrem Webbrowser diese Datei auf http://localhost/php_info.php3. Erscheinen daraufhin umfängliche Tabellen mit Informationen zur PHP-Version, zu installierten Extensions

usw.? Dann ist PHP bereits installiert! Findet sich in der Tabelle "Extensions" auch ein Eintrag MySQL mit allerlei Angaben? Dann unterstützt Ihre PHP-Installation auch MySQL! Wenn nicht, nun dann sollten Sie vielleicht erwägen, zunächst eine weitere Installationsprozedur über sich ergehen zu lassen (Hilfestellung zur Installation von PHP gibt Kapitel 13). Falls Sie nicht genau wissen, wo Sie Ihr Document Root Verzeichnis finden können: Versuchen Sie es einmal mit dem Befehl `locate htdocs` (unter Apache hat Document Root standardmäßig den Namen `htdocs`): übliche Installationspfade sind z. B.:

```
/usr/local/apache/htdocs  
/usr/local/httpd/htdocs  
/usr/local/www/htdocs
```

Falls `locate` Ihnen zeigt, daß in Ihrem System mehrere `htdocs`-Verzeichnisse existieren, probieren Sie vorsichtshalber alle aus.

phpMyAdmin ist häufig die einzige Möglichkeit, an seine Datenbank bei einem Massen-ISP heranzukommen: In der Regel erlauben Ihnen die Provider mit den günstigen „Profi-Paketen“ keinen telnet-Zugriff auf den Server, auf dem die versprochene MySQL-Datenbank gehostet ist. Es besteht daher normalerweise keine Möglichkeit, einen MySQL-Client auf dem Server des Provider zu starten. Provider, die Ihnen MySQL anbieten, bieten Ihnen in aller Regel aber auch PHP an (wenn nicht, dann wird es spätestens jetzt Zeit, den Provider zu wechseln). Da phpMyAdmin lediglich aus einer Reihe von HTML- und PHP-Dateien besteht (und auch der billigste ISP Ihnen die Möglichkeit einräumen muß, HTML-Dateien – in der Regel via FTP – in ein Verzeichnis auf dem Webserver aufzuspielen), können Sie diese Dateien problemlos zunächst zu Hause lokal konfigurieren und dann in Ihr Verzeichnis beim Provider aufspielen. Anschließend rufen Sie über Ihren Webbrowser die gerade nach <http://www.ihre-homepage.de/phpMyAdmin/> kopierten Dateien auf und administrieren Ihre MySQL-Datenbank.

Da es unseres Wissens derzeit keine deutschsprachige Dokumentation zu phpMyAdmin gibt und – wie ausgeführt – phpMyAdmin häufig die eierlegende Wollmilchsau für den ambitionierten Website-Hobbyisten darstellt, fällt die Darstellung etwas ausführlicher aus.

Es empfiehlt sich, die Konfiguration von phpMyAdmin zunächst an einem „geschützten“, von außen nicht zugreifbaren Ort vorzubereiten, und erst die fertig konfigurierte Installation nach Document Root zu kopieren (Document Root sei in unserem Beispiel `/usr/local/www/htdocs`). Zunächst besorgt man sich die Source-Distribution und entpackt sie:

```
> cd /usr/local/src
> # entweder mit cp von CD oder mit wget aus dem Internet
> # die Source holen
> cp /cdrom/phpMyAdmin/phpMyAdmin_2_0_5_tar.gz . # oder
> wget http://www.phpwizard.net/phpMyAdmin/\
>   phpMyAdmin_2.0.5.tar.gz
> gunzip < phpMyAdmin_2.0.5.tar.gz | tar xvf -
> cd phpMyAdmin
```

Der nächste Schritt ist die Anpassung von phpMyAdmin an die lokalen Verhältnisse. Hierzu dient die Datei `config.inc.php3`. In dieser müssen Sie einer Reihe von Variablen Werte zuweisen. Diese Variablen werden im Folgenden besprochen:

phpMyAdmin Variable	Beschreibung
<code>\$cfgServers</code> array z. B.: <code>\$cfgServers[1]['host']</code> <code>\$cfgServers[2]['host']</code> <code>\$cfgServers[3]['host']</code>	phpMyAdmin unterstützt die Administration mehrerer MySQL-Server. Aus diesem Grund wird ein zweidimensionales <code>\$cfgServers</code> -array zur Verfügung gestellt, mit dessen Hilfe unterschiedliche Logininformationen für verschiedene Server angegeben werden können. In <code>\$cfgServers[1]['host']</code> wird der Host-

<pre>\$cfgServers[1] ['adv_auth'] \$cfgServers[2] ['adv_auth'] \$cfgServers[3] ['adv_auth'] ...</pre>	<p>name des ersten Servers abgelegt, in <code>\$cfgServers[2]['host']</code> der Hostname des zweiten Servers usw. Wenn Sie auf Datenbanken lediglich eines Servers zugreifen wollen, lassen Sie einfach die Hostname-Einträge für die anderen beiden in der Konfigurationsdatei vorgegebenen Server frei.</p>
<pre>\$cfgServers[n] ['port'] string</pre>	<p>Die Port Nummer des n-ten Servers. Sie können den Eintrag leer lassen (schreiben Sie: <code>\$cfgServers[n]['port'] = ''</code>), wenn Sie mit Standardwerten arbeiten wollen. Beispiel: <code>\$cfgServers[1]['port'] = '3306'</code>;</p>
<pre>\$cfgServers[n] ['host'] string</pre>	<p>Der Hostname des n-ten MySQL-server. Wenn Sie phpMyAdmin auf dem Rechner installieren, auf dem auch der MySQL-Server installiert ist, geben Sie „localhost“ an. Beispiel: <code>\$cfgServers[1]['host'] = 'localhost'</code>;</p>
<pre>\$cfgServers[n] ['adv_auth'] boolean</pre>	<p>Hier bestimmen Sie, ob Sie die Standard- ('basic') oder die erweiterte ('advanced') Authentikation verwenden wollen. 'basic'-Authentikation (<code>\$cfgServers[1]['adv_auth'] = false;</code>) ist der einfachere aber weniger sichere Weg: Benutzername und Paßwort werden in der Datei <code>config.inc.php3</code> gespeichert (die in Document Root liegt, also tendenziell gefährdet ist, von außen angegriffen zu werden). Bei der erweiterten Authentikation werden nur Logininformationen eines Standardusers gespeichert (der nichts dürfen sollte, außer die Authentikationsinformationen des MySQL-DBMS zu lesen!). Über eine HTTP-Authentikation (das ist der Standard-Benutzername/Kennwort-Dialog, den die Webbrowser in einem solchen Fall dem Benutzer vorlegen) werden sodann die Login-Informationen vom Nutzer erfragt. Diese Angaben werden mit den Angaben in</p>

	<p>der MySQL-Nutzerverwaltung abgeglichen. Zugriff erhalten nur berechtigte MySQL-Benutzer (und diese auch nur auf die Datenbanken, für die sie eine Berechtigung haben). Die erweiterte Authentikation ist relativ sicher, weil MySQL-Kennwörter nicht ohne Weiteres entschlüsselt werden können (sofern nicht eine der Versionen 3.22.26a bis 3.23.12 installiert ist; diese enthalten ein Sicherheitsloch) und der Standarduser nur lesenden Zugriff auf die Benutzerverwaltung hat (die Datenbank <code>mysql</code> des MySQL-DBMS).</p> <p>Beispiel: <code>\$cfgServers[1]['adv_auth'] = true;</code></p>
<p>\$cfgServers[n] ['user'] \$cfgServers[n]['password'] string</p>	<p>Das Benutzer/Kennwort-Paar, mit dem phpMyAdmin bei der Standard-Authentikation (<code>\$cfgServers[1]['adv_auth'] = false;</code>) auf die Datenbank zugreift. Wenn die erweiterte Authentikation verwendet werden soll, bleiben diese Variablen leer.</p> <p>Beispiel: <code>\$cfgServers[1]['user'] = 'DB-Admin';</code> <code>\$cfgServers[1]['password'] = 'top-secret';</code></p>
<p>\$cfgServers[n]['stduser'] \$cfgServers[n]['stdpass'] string</p>	<p>Das Benutzer/Kennwort-Paar des Standardbenutzers (der <i>keinen</i> Zugriff auf Nutzerdatenbanken hat bzw. haben sollte!), mit dem phpMyAdmin bei der erweiterten Authentikation (<code>\$cfgServers[1]['adv_auth'] = true;</code>) auf die Benutzerverwaltung von MySQL zugreift, um die hernach per HTTP-Request angeforderte Benutzer/Kennwort-Kombination für den Datenbankzugriff zu verifizieren. Wenn die Standard-Authentikation verwendet werden soll, bleiben diese Variablen leer.</p> <p>Beispiel: <code>\$cfgServers[1]['stduser'] = 'Nobody';</code> <code>\$cfgServers[1]['stdpass'] = 'nothing';</code></p>
\$cfgServers[n]	Wenn Sie hier einen Datenbanknamen angeben, ist

<p>['only_db'] string</p>	<p>dies die einzige Datenbank, die die Nutzer zu Gesicht bekommen. Beispiel: <code>\$cfgServers[1]['only_db'] = 'oeffentliche_db';</code></p>
<p>\$cfgServers[n]['verbose'] string</p>	<p>Wenn Sie phpMyAdmin mit mehreren Datenbankservern benutzen (Sie also mehr als einen <code>\$cfgServers[n]['host']</code>-Eintrag vorgenommen haben), wählen Sie später in der Benutzeroberfläche von phpMyAdmin den zu verwendenden Server in einem Pull-down-Menü aus. Hier können Sie den Namen angeben, der in diesem Menü erscheinen soll (geben Sie nichts an, wird der Hostname verwendet). Beispiel: <code>\$cfgServers[1]['verbose'] = 'Interner Testserver';</code></p>
<p>\$cfgServerDefault integer</p>	<p>Hier wird der Array-Index des Datenbankservers angegeben, zudem phpMyAdmin beim Start automatisch Verbindung aufnehmen soll. Wenn nur ein Server konfiguriert wurde (das ist der Regelfall) <i>muß</i> der Wert „1“ zugewiesen werden. Wurden mehrere Datenbankserver konfiguriert, können (neben den Array-Indizes der anderen DB-Server) auch „0“ angegeben werden; in diesem Fall wird in der Benutzeroberfläche von phpMyAdmin lediglich ein Pull-down-Menü der möglichen Server zur Auswahl vorgelegt, ein automatische Verbindungsaufnahme unterbleibt. Beispiel: <code>\$cfgServerDefault = 1;</code></p>
<p>\$cfgManualBase string</p>	<p>Wenn hier ein URL zur MySQL-Dokumentation angegeben wird, werden in der phpMyAdmin-Benutzeroberfläche - kontextsensitiv - Hilfe-Links eingeblendet, die zum passenden Abschnitt der Dokumentation führen. Voreingestellt ist http://www.tcx.se/Manual_chapter (Dokumentation auf der MySQL-Homepage).</p>

	<p>Wenn Sie sich mit wget http://www.mysql.com/Manual/html-manual.tar.gz die Dokumentation „saugen“ und auf Ihrem Webserver z. B. unter <Document Root>/intern/doku/MySQL- Doku einstellen, können Sie von phpMyAdmin aus auf diese lokale Version der Dokumentation zugrei- fen. Geben Sie dann an: Beispiel: \$cfgManualBase = 'http://localhost/intern/doku/MySQL-Doku';</p>
\$cfgConfirm boolean	<p>Falls Sie diese ständigen „Sind-Sie-wirklich-sicher,- daß...“-Warnungen nicht mehr sehen können, im- mer wenn Sie gerade dabei sind, Ihre Datenbanken oder Tabellen zu löschen, dann können Sie diese hier abstellen. Im folgenden Beispiel (dies ist auch die Voreinstellung) sind die Warnungen eingeschalt- tet. Beispiel: \$cfgConfirm = true;</p>
\$cfgPersistent- Connections boolean	<p>Sollen Persistente Datenbankverbindungen aufge- baut werden (Abfragen mit der php-Funktion mysql_pconnect() anstatt mysql_connect())? Diese Einstellung ist nur relevant, wenn PHP als Apache- Modul installiert ist, bei der CGI-Version unterschei- den sich die beiden Funktionen nicht (vgl. Kap. 13). Persistente Verbindungen sind etwas schneller, weil nicht ständig Netzwerkverbindungen geöffnet und geschlossen werden müssen. Da MySQL (im Gegen- satz z. B. zu Oracle) beim Verbindungsaufbau aber sehr schnell ist und andererseits sehr viele offene Verbindungen schon einmal den Apache in die Knie zwingen können, empfiehlt es sich, bei der konser- vativen Voreinstellung (siehe Beispiel) zu bleiben. Beispiel: \$cfgPersistentConnections = false;</p>
\$cfgMaxRows	<p>Anzahl der Reihen, die angezeigt werden, wenn Sie</p>

integer	durch die Ergebnistabellen Ihrer SQL-Abfragen browsen. Wenn die Ergebnistabelle mehr als \$cfgMaxRows Zeilen enthält, werden Vor/Zurück-Links ausgegeben. Voreinstellung: 30. Beispiel: \$cfgMaxRows = 30;
\$cfgMaxInputsize string	Größe des Eingabefeldes, mit dem eine neue Zeile in eine Tabelle eingefügt werden kann. Voreinstellung: 300px (Pixel). Beispiel: \$cfgMaxInputsize = '300px';
\$cfgBorder string	Die „Dicke“ der Tabellenränder in phpMyAdmin. Beispiel/Voreinstellung: \$cfgBorder = '0';
\$cfgThBgcolor \$cfgBgcolorOne \$cfgBgcolorTwo string [HTML color]	Die Farben, die für die Tabellenüberschriften und die Tabellenzeilen verwendet werden. Wenn Ihnen die Voreinstellung (hellblau "#D3DCE3", hellgrau "#CCCCCC" und dunkelgrau "#DDDDDD") zu konservativ sind versuchen Sie es doch einmal mit einer sonnigeren Variante: Beispiel: \$cfgThBgcolor = "#FFCC66"; \$cfgBgcolorOne = "#FFFF99"; \$cfgBgcolorTwo = "#FFFFCC";
\$cfgOrder string ["DESC"]["ASC"]	Definiert, ob Felder in aufsteigender (ascending, "ASC") oder in absteigender (descending, "DESC") dargestellt werden sollen, wenn auf einen Feldnamen geklickt wird. Beispiel/Voreinstellung: \$cfgOrder = "ASC";
\$cfgShowBlob boolean	Bestimmt, ob BLOB-Felder gezeigt werden, wenn der Inhalt einer Tabelle betrachtet wird. Da BLOB-Felder Binärdaten enthalten können (z. B. Bilddateien), kann eine Darstellung dieser Felder zu Problemen führen. In der Regel enthalten BLOB-Felder allerdings nur längere Texte, so daß eine Darstellung meistens erwünscht ist.

	Beispiel/Voreinstellung: <code>\$cfgShowBlob = true;</code>
\$cfgShowSQL boolean	<p>Hier können Sie einstellen, ob die SQL-Abfragen, die phpMyAdmin auf Grund Ihrer Klicks und Eintragungen generiert, angezeigt werden sollen, oder nicht. Wenn Sie noch kein Datenbankprofi sind, können Sie hieraus viel über die Verwendung von SQL-Befehlen lernen!</p> <p>Beispiel/Voreinstellung: <code>\$cfgShowSQL = true;</code></p>
\$cfgColumnTypes \$cfgAttributeTypes \$cfgFunctions array	<p>Definiert alle zulässigen Typen einer MySQL-Spalte (<code>\$cfgColumnTypes</code>) bzw. alle zulässigen Attribute eines Felds (<code>\$cfgAttributeTypes</code>). Nur auf die hier gelisteten Einträge kann später unter der phpMyAdmin-Oberfläche zugegriffen werden. In der Regel besteht für Änderungen allerdings keine Notwendigkeit (besorgen Sie sich ggf. lieber eine neue Version von phpMyAdmin). Das Array <code>\$cfgFunctions</code> enthält alle von MySQL unterstützten Funktionen. Wenn Sie UDFs (User Defined Functions) erstellen, müssen Sie diese hier eintragen, wenn Sie über phpMyAdmin auf Ihre Funktionen zugreifen wollen.</p> <p>Beispiel: <code>\$cfgFunctions = array("ASCII", ..., "RAND", "MEINE_UDF");</code></p>
Funktion require("Sprachdatei");	<p>In der Datei <code>config.inc.php3</code> finden Sie auch eine Zeile, in der die PHP-Funktion <code>require()</code> aufgerufen wird. Mit dieser Funktion werden in diesem Fall Sprachanpassungsdateien aufgerufen. Sie können sich von phpMyAdmin in elf verschiedenen Sprachen unterstützen lassen (Chinesisch, Deutsch, Dänisch, Englisch, Französisch, Italienisch, Katalanisch, Norwegisch, Portugiesisch, Russisch, Spanisch)! Voreingestellt ist Englisch.</p> <p>Beispiel: <code>require("german.inc.php3");</code></p>

Ausgestattet mit den Informationen aus dieser Tabelle sollte die Konfiguration von phpMyAdmin nun keine größeren Schwierigkeiten mehr bereiten. Der schwierigste Teil bleibt die Einstellung des Authentikationssystems. Dies liegt weniger an phpMyAdmin (in phpMyAdmin ist diese Frage auf für den Nutzer transparente und einfache Weise gelöst), sondern mehr an der schwierigeren Authentikations-Konfiguration des Apache Webservers und an der – unserer Meinung nach komplizierten und nicht-intuitiven – Benutzerverwaltung von MySQL. Dieser entscheidende Punkt – wenn hier etwas schief läuft, können Sie nicht auf die Datenbank zugreifen – wird im Folgenden deshalb detailliert besprochen.

Angenommen, es soll nur eine Standard-Authentikation durchgeführt werden (wovon wir abraten, wenn Sie ans Internet angeschlossen sind oder Sie in einer Multiuser-Umgebung arbeiten). Dann muß vor der Konfiguration von phpMyAdmin unter MySQL mindestens ein Benutzer eingerichtet sein (und zwar ein Datenbankadministrator, der Vollzugriff auf MySQL haben soll; dies ist bei korrekter Installation immer gegeben). Wenn Sie in der Benutzerverwaltung noch keine Änderungen vorgenommen haben (aber MySQL erfolgreich installiert), dann wurde durch den Aufruf des Skripts `mysql_install_db` zwei Benutzer angelegt: `root` (der Datenbankadministrator, der darf alles!) und der Namenlose Nutzer (der alles mit Datenbanken machen darf, die `test` heißen oder deren Name mit `test_` beginnt, darüberhinaus aber keinerlei Rechte hat). Sofern nicht schon geschehen, sollte spätestens jetzt den `root`-Account zu MySQL mit einem Kennwort geschützt werden. Das geht z. B. so:

```
shell> mysql -u root mysql
mysql> UPDATE user SET Password=PASSWORD('top_secret')
mysql> WHERE user='root';
mysql> FLUSH PRIVILEGES;
```

der Befehl `mysql ...` ruft den MySQL-Client als User `root` auf und öffnet die Datenbank `mysql`. Wenn das so nicht klappt, haben Sie wahrscheinlich bereits ein Paßwort vergeben. Mit `UPDATE ...` wird ein neues Paßwort gesetzt. Der Befehl `FLUSH PRIVILEGES` wird gerne vergessen und dann bleiben alle Änderungen wirkungslos. Modifikationen in der Benutzerverwaltung müssen dem System mit diesem Befehl zur Kenntnis gebracht werden. Von jetzt an funktioniert `mysql -u root` nicht mehr, Sie erhalten die Fehlermeldung

```
ERROR 1045: Access denied for user: 'root@localhost'  
(Using password: NO)
```

Sie müssen künftig `mysql -u root -p` verwenden; dann werden Sie nach dem Paßwort gefragt. Nach dieser Vorarbeit konfigurieren Sie phpMyAdmin in der Datei `config.inc.php3` wie folgt:

```
$cfgServers[1]['host'] = 'localhost';  
$cfgServers[1]['port'] = '';  
$cfgServers[1]['adv_auth'] = false;  
$cfgServers[1]['stduser'] = '';  
$cfgServers[1]['stdpass'] = '';  
$cfgServers[1]['user'] = 'root';  
$cfgServers[1]['password'] = 'top_secret';  
$cfgServers[1]['only_db'] = '';  
$cfgServers[1]['verbose'] = '';
```

Bevor Sie jetzt das phpMyAdmin-Verzeichnis nach Document Root kopieren, sollten Sie aus Sicherheitsgründen das Zielverzeichnis auf jeden Fall mit einem Kennwortschutz versehen. Wenn Sie Vollzugriff auf den Rechner haben, auf dem der Apache Webserver installiert ist, editieren Sie einfach die Apache-Konfigurationsdatei. Das ist in neueren Installationen die Datei `httpd.conf`, in älteren Installationen kann es auch die Datei `access.conf` sein. Sie finden Sie in der Regel unter `/etc/httpd/httpd.conf` oder `/etc/httpd.conf` oder `/usr/local/apache/conf/httpd.conf` (im Zweifelsfall schafft ein

locate httpd.conf Klarheit). In die httpd.conf fügen Sie folgendes ein:

```
<Location /intern/phpMyAdmin>
AuthType Basic
AuthUserFile /usr/local/apache/conf/passwd
AuthName "phpMyAdmin Datenbankadministration"
Require valid-user
</Location>
```

Anschließend müssen Sie die Konfigurationsdateien des Apache neu einlesen, z. B. mit:

```
> /usr/local/apache/bin/apachectl graceful
```

Nun ist das Verzeichnis `/usr/local/www/htdocs/intern/phpMyAdmin` (wenn `/usr/local/www/htdocs` Document Root ist) gegen Zugriffe von außen so gut geschützt, daß Sie selbst nicht mehr herankommen. Um auf das Verzeichnis zugreifen zu können, müssen Sie zunächst die Datei `/usr/local/apache/conf/passwd` und mindestens einen Benutzer anlegen. Dies geschieht mit dem Befehl `htpasswd`, wobei beim ersten Aufruf des Programms die Option `-c` angegeben werden muß, damit eine neue Paßwortdatei angelegt wird.

```
> cd /usr/local/apache/bin
> ./htpasswd -c /usr/local/apache/conf/passwd dbadmin
> New password: *****
> Re-type new password: *****
> Adding password for user dbadmin
```

Ab sofort kann auf das Verzeichnis `http://localhost/intern/phpMyAdmin` nur noch zugreifen, wer sich als `dbadmin` einloggt und das eben vergebene Kennwort weiß. Sie sollten aber wissen, das `AuthType Basic` nicht sehr sicher ist (immerhin reicht es in der Regel, um Windows-gebildete "Power"-User fernzuhalten), weil die Paßwörter unverschlüsselt übertragen werden und da-

her belauscht werden können. Wenn Sie möchten, können Sie nun weiteren Nutzern den Zugriff auf das Verzeichnis gestatten, indem Sie der Paßwortdatei nach dem oben beschriebenen Verfahren weitere Benutzer hinzufügen.

Auch wenn Sie keinen Vollzugriff auf das System haben, ist möglicherweise nicht alles verloren, jedenfalls dann nicht, wenn die Apache-Konfiguration lokale `.htaccess`-Dateien zuläßt. Diese erlaubt der Apache Webserver in der Standardkonfiguration (man kann das allerdings mit der Apache Direktive `AllowOverride` insgesamt oder für einzelne Verzeichnisse ganz oder teilweise unterbinden). Probieren Sie es im Zweifelsfall einfach aus, indem Sie im zu schützenden Verzeichnis eine Datei `.htaccess` erzeugen (der Punkt am Anfang ist erforderlich!) und in diese die oben beschriebene Befehlsfolge (`<Location ...`) einfügen. Die Pfade zur Paßwortdatei werden Sie entsprechend ändern müssen; wenn irgend möglich sollte die Paßwortdatei nicht unter Document Root abgelegt werden.

Alles ist einfacher, sollten Sie sich - wie wir es Ihnen empfohlen haben - zu einer "advanced" Authentikation entschließen. Die Funktionsweise ist hier anders: statt an ungeschützter Stelle (in öffentlich zugänglichen Web-Verzeichnissen!) ein Paßwort für den Datenbankadministrator abzulegen, wird - öffentlich - lediglich ein Paßwort für einen Nutzer abgelegt (nennen wir ihn "Niemand"), der keinerlei Rechte an irgendeiner Datenbank hat, aber berechtigt ist, die Benutzerverwaltung von MySQL zu konsultieren (d. h. Leserechte für die Verwaltungsdatenbank `mysql` hat). Will nun ein Benutzer auf das phpMyAdmin-Verzeichnis via HTTP zugreifen, verlangt phpMyAdmin vom Einlaß Begehrenden zunächst (via Standard HTTP-Authentikation) eine Benutzer/Kennwort-Kombination. Mit "Niemand"s Hilfe überzeugt sich phpMyAdmin nun davon, ob dem Begehren stattzugeben oder es abzulehnen ist. Zu diesem Zwecke vergleicht phpMyAdim die Login-Informationen mit den Angaben in der Benutzerverwaltung von MySQL.

Wir gehen wieder davon aus, daß MySQL erfolgreich installiert und das Initialisierungsskript `mysql_install_db` ausgeführt wurde, aber sonst keine weiteren Maßnahmen bezüglich der Benutzerverwaltung ergriffen haben. Dann sollten Sie zunächst (wie oben schon einmal beschrieben) dem Datenbankadministrator `root` (der übrigens nicht verwandt oder verschwägert ist mit dem Linux-Systemadministrator `root` und insbesondere auch mit dessen Paßwort rein gar nichts zu tun hat) ein Paßwort zuweisen und so dieses scheunentorgroße Sicherheitsloch schleunigst schließen (siehe UPDATE-Befehl im nachfolgenden Listing). Anschließend richten Sie einen User "Niemand" ein (siehe 1. INSERT-Befehl), dem Sie alles verbieten (durch die vielen "N"s, die NO! bedeuten, genaueres zur Nutzerverwaltung lesen Sie bitte im Kapitel 9.2). Nur die Datenbank `mysql` (gemeint ist hier die Verwaltungsdatenbank `mysql` des DBMS MySQL) darf er sich anschauen (dafür sorgt das "Y"). Dieses eine kleine Privileg ist notwendig, damit phpMyAdmin unter Nutzung des Users "Niemand" nachschauen kann, ob die Login-Informationen korrekt sind, 2. INSERT-Befehl). Zu guter Letzt lassen wir MySQL die Authentikationsinformationen mit `FLUSH PRIVILEGES` neu einlesen (nicht vergessen!).

```
shell> mysql -u root
mysql> use mysql;
mysql> UPDATE user SET Password=PASSWORD('top_secret')
WHERE user='root';
mysql> INSERT INTO user VALUES('localhost', 'Niemand',
PASSWORD('nothing'), 'N', 'N', 'N', 'N', 'N',
'N', 'N', 'N', 'N', 'N', 'N', 'N', 'N', 'N');
mysql> INSERT INTO db VALUES('localhost', 'mysql',
'Niemand', 'Y', 'N', 'N', 'N', 'N', 'N', 'N',
'N', 'N', 'N');
mysql> FLUSH PRIVILEGES;
```

Denken Sie daran, daß Sie nach dieser Operation den MySQL-Client mit `mysql -u root -p` starten müssen, damit Sie kein "Access denied" wegen fehlenden Paßworts erhalten. Der Rest ist einfach. Zur Konfiguration von phpMyAdmin mit erweiterter Authentifikation nehmen Sie die folgenden Eintragungen in der Datei `config.inc.php3` vor:

```
$cfgServers[1]['host'] = 'localhost';
$cfgServers[1]['port'] = '';
$cfgServers[1]['adv_auth'] = true;
$cfgServers[1]['stduser'] = 'Niemand';
$cfgServers[1]['stdpass'] = 'nothing';
$cfgServers[1]['user'] = '';
$cfgServers[1]['password'] = '';
$cfgServers[1]['only_db'] = '';
$cfgServers[1]['verbose'] = '';
```

Wenn Sie möchten, können Sie jetzt noch den Dokumentationspfad auf Ihre lokale Quelle setzen (sofern Sie die MySQL-Dokumentation heruntergeladen bzw. von CD kopiert haben), die Sprache auf Deutsch setzen und ein wenig mit den Farbe der Tabellen in phpMyAdmin spielen:

```
$cfgManualBase = 'http://localhost/intern/doku/MySQL-Doku';
require ('german.inc.php3')
$cfgThBgcolor = "#FFCC66";
$cfgBgcolorOne = "#FFFF99";
$cfgBgcolorTwo = "#FFFFCC";
```

Nachdem diese – doch etwas aufwendigere – Konfigurationsprozedur durchgeführt wurde, kann das phpMyAdmin-Verzeichnis nach Document Root verschoben werden und die Datenbanken via phpMyAdmin administriert werden (durch Aufruf des URL: <http://localhost/intern/phpMyAdmin>). Wenn alles geklappt hat (wovon wir ausgehen) zeigt sich Ihnen phpMyAdmin ähnlich wie in Abb. 9.6 wiedergegeben:

mysql006.tif/psd

Abb. 9.6 phpMyAdmin: Der Begrüßungsbildschirm von phpMyAdmin zeigt sich in schlichtem Charme. phpMyAdmin ist ein Tiefstapler: Der Funktionsumfang des Programms läßt kaum Wünsche offen!

In einer Multiuser-Umgebung hat die Verwendung der advanced Authentikation von phpMyAdmin neben der wesentlich höheren Sicherheit einen weiteren gravierenden Vorteil: Soll Anwendern Zugriff auf die MySQL-Datenbank via phpMyAdmin gewährt werden, brauchen Sie keine zusätzlichen Authentikationsmechanismen zu bemühen, wie z. B. die oben Beschriebenen des Web-Servers Apache. Die Benutzerverwaltung von MySQL steuert nun gleichzeitig auch die Zugriffsmöglichkeiten über phpMyAdmin. Sollten Sie z. B. unter MySQL einen Benutzer einrichten, der nur die Datenbank "Spielwiese" benutzen darf, dann bekommt dieser Nutzer, nachdem er sich gegenüber phpMyAdmin authentiziert hat, auch nur die Datenbank "Spielwiese" zu Gesicht. Alle anderen möglicherweise existierenden Datenbanken kann dieser Nutzer weder sehen noch verändern (Wenn Ihnen jetzt noch nicht ganz klar ist, wie Sie das hin bekommen, dann macht das nichts: Kapitel 9.2 behandelt das Thema noch einmal gründlicher).

Noch ein Tipp zum Schluß: Versuchen Sie nicht – nach der Devise: Doppelt gemoppelt hält besser – sowohl die erweiterte Authentikation von phpMyAdmin als auch gleichzeitig die Authentikationsmechanismen des Apache zu nutzen. Die beiden Mechanismen kommen sich ins Gehege, so daß überhaupt nicht mehr auf phpMyAdmin zugegriffen werden kann.

Der schlichte Charme des Begrüßungsbildschirm von phpMyAdmin verdeckt die Möglichkeiten des Programms: Der Funktionsumfang des Tools läßt kaum Wünsche offen. Im linken Frame des Browsers werden alle Datenbanken aufgelistet, für die dem Benutzer, der sich zuvor authentiziert hat, mindestens "SELECT"-Rechte eingeräumt wurden. Dem Datenbankadministra-

tor `dbadmin` (sofern Sie diesen wie oben beschrieben eingerichtet haben) werden alle Datenbanken angezeigt. Bei allen Operationen wird (von MySQL) stets geprüft, ob der Benutzer über die entsprechenden Rechte verfügt, wenn nicht, wird die Operation mit einer Fehlermeldung abgebrochen. An vielen Stellen von phpMyAdmin finden Sie einen Link [DOKUMENTATION]; dieser führt Sie - kontextsensitiv - zum entsprechenden Abschnitt des MySQL-Handbuchs. Der Klick auf den [DOKUMENTATION]-Link hinter `MYSQL-LAUFZEIT-INFORMATIONEN ANZEIGEN` führt z. B. zum Abschnitt "7.20 SHOW syntax (Get information about tables, columns...)" des Handbuchs. Der `MySQL-SHOW`-Befehl ist es nämlich, der von phpMyAdmin aufgerufen wird, um die `MySQL-Laufzeit-Informationen` zu erhalten.

mysql007.tif/psd

Abb. 9.7: phpMyAdmin informiert: die Ausgaben der Befehle "MySQL-Laufzeit-Informationen anzeigen", "MySQL-System-Variablen anzeigen", "Prozesse anzeigen" liefern wichtige Informationen zur Konfiguration und zum aktuellen Zustand der Datenbank.

Für diejenigen von uns, die noch nicht fließend SQL sprechen, sind die Ausgaben in den Feldern "SQL-Befehl" interessant. Dort steht z. B. nach Klick auf `MYSQL-LAUFZEIT-INFORMATIONEN ANZEIGEN`:

```
SHOW STATUS LIMIT 0,30
```

Dies ist exakt das SQL-Statement, daß angegeben werden muß, um die in Abb. 9.7 links dargestellte Ausgabe zu produzieren. Die einzelnen in der Abbildung ausschnitthaft ausgewiesenen Informationen sollen an dieser Stelle nicht besprochen werden, genaueres findet sich im MySQL-Handbuch (unter http://www.mysql.com/Manual_chapter/manual_Reference.html#SHOW).

Wichtig sind diese Angaben immer dann, wenn etwas nicht klappt oder Sie Geschwindigkeitsoptimierungen vornehmen wollen oder müssen. Interessant ist in manchen Fällen die Ausgabe des Befehls `PROZESSE ANZEIGEN`: Im Beispiel wird unter ID 104 ein Prozeß des Users "boss" ausgewiesen, der sich von local-

host aus eingelogged hat und seit 9 Sekunden schläft (es handelte sich um einen Zugriff über den MySQL-Client `mysql` unter Linux). Mit dem KILL-Befehl am Ende der Zeile können Sie den Prozeß abschießen (und damit vielleicht `boss` wecken?). Außerdem sehen Sie einen Prozeß des Benutzers "niemand" (unter dieser Benutzerkennung hat phpMyAdmin nachgeschaut, zu welchen Rechten die Benutzer/Kennwortkombination des Benutzers `dbadmin` verhilft), sowie einen Prozeß des Benutzers `dbadmin` (der offenbar von "niemand" erfolgreich authentiziert werden konnte) und sich gerade die Prozeßliste anschaut.

mysql008.tif/psd

Abb. 9.8: phpMyAdmin Datenbank-Bildschirm: Im Datenbankbildschirm können Sie auf alle Tabellen der Datenbank zugreifen, beliebige SQL-Befehle ausführen, Sicherheitskopien der gesamten Datenbank anlegen und vieles mehr.

Klickt man im linken Frame des phpMyAdmin-Begrüßungs-Bildschirms (Abb. 9.6) auf eine der angezeigten Datenbanken (z. B. `mysql`, die Datenbank, die in MySQL der Benutzerverwaltung dient) öffnet sich Ihnen der phpMyAdmin Datenbankbildschirm (Abb. 9.8). Von hier aus können Sie auf alle Tabellen der Datenbank zugreifen (1), z. B. um sich den Inhalt einer Tabelle anzeigen zu lassen (klicken Sie dazu in der Zeile `user` auf ANZEIGEN) . Bei großen Tabellen wollen Sie vielleicht nicht alle Spalten sehen, dann verwenden Sie den Menüpunkt TEILW. ANZEIGEN. Der sich daraufhin öffnende Bildschirm erlaubt die Auswahl bzw. Abwahl einzelner Felder. Wenn Sie wollen, können Sie auch gleich noch inhaltliche Kriterien angeben ("Zeige mir nur Tabellenzeilen, die dieses oder jenes enthalten"). Wollen Sie einen neuen Datensatz anlegen (in die Tabelle eine neue Zeile einfügen) wählen Sie den Menüpunkts EINFÜGEN. Der sich daraufhin öffnende Bildschirm stellt Ihnen die notwendigen Eingabefelder oder Dropdown-Menüs zur Verfügung. Die Menüpunkte LÖSCHEN bzw. LEEREN entfernen eine Tabelle entweder vollständig aus der Datenbank (LÖSCHEN) bzw. entfernen alle Einträge aus einer Tabelle (LEEREN). Sofern Sie bei der Konfiguration

von phpMyAdmin die Voreinstellungen belassen haben (`$cfg-Confirm = true`), brauchen Sie übrigens nicht befürchten, durch einem unvorsichtigen Klick Ihren Tabellen oder Datenbanken zu verlieren. Die Antwort auf die Frage:

Möchten Sie wirklich diese Abfrage ausführen:
`DROP DATABASE xx? #` oder
`DROP TABLE xx?`

sollten Sie sich allerdings gut überlegen. Wenn Sie jetzt "ja" sagen, dann ist die Datenbank oder die Tabelle wirklich und unwiderruflich futsch, es sei denn, die Vorschläge zur Datensicherung in Kapitel 9.3 wurden befolgt. Mit Hilfe des Menüpunkts Eigenschaften können Sie die Eigenschaften einer Tabelle verändern; genaueres folgt wenige Absätze später.

Sie können natürlich nicht nur bestehende Tabellen bearbeiten sondern auch neue anlegen (2). Nachdem Sie einen Namen für die Tabelle und die Anzahl der Spalten angegeben haben führt der Klick auf `OK` Sie zu einem weiteren Bildschirm, indem Sie für jede Spalte (jedes Feld) den Datentyp auswählen können (Zahl, Text, Datum usw.), ggf. eine Länge angeben (z. B. 30 Zeichen), Attribute setzen (z. B. Zahlen ohne Vorzeichen) und festlegen können, ob ein Feld den Wert `NULL` besitzen darf. Sie können ein Feld mit einem Wert vorbelegen (der `DEFAULT`-Wert einer Spalte wird immer dann verwendet, wenn eine Zeile angelegt wird *ohne* der betreffenden Spalte einen Wert zuzuweisen. In der Tabelle `user` der Datenbank `mysql` sind z. B. alle "Privileg"-Felder mit "N", `NO!`, vorbelegt. Standardmäßig darf jeder Benutzer also erst einmal nichts!). Sie können einer Spalte "Extra"-Fähigkeiten zuweisen (derzeit nur "auto_increment") und Indizes sowie Primärschlüssel anlegen.

Für die schnelle Abfrage zwischendurch (aber natürlich auch für jede längere) können Sie das Eingabefeld (3) `SQL-BEFEHL(E) AUSFÜHREN IN DATENBANK XY` verwenden (Wenn Sie Ihre Befehle stattdessen in einem Editor Ihrer Wahl schreiben - bei mehreren Be-

fehlen die Semikola nicht vergessen – und die Befehlssequenzen mit Copy und Paste in das Eingabefeld einfügen, haben Sie eine History ihrer SQL-Versuche, so daß Sie komplexe Abfragen abspeichern und später wiederverwenden können). Sie können aber auch gleich eine ASCII-Datei mit (My)SQL-Befehlen in der Dateiauswahlbox auswählen, die erscheint wenn Sie auf DURCHSUCHEN ... klicken. Die Befehle in dieser Datei werden prompt ausgeführt (sofern keine Syntaxfehler enthalten sind).

Dies ist übrigens eine der Möglichkeiten, mit denen Sie eine Sicherheitskopie Ihrer Datenbank wieder aufspielen können; wenn Ihre Datenbank bei einem Massen-ISP gehostet wird, ist es möglicherweise Ihre einzige Möglichkeit. Sie können nämlich einfach einen "Dump" Ihrer Datenbank erzeugen (5): Diesen können Sie sich entweder am Bildschirm anzeigen lassen, oder – wenn sie SENDEN auswählen – auch auf die Festplatte des Rechners downloaden, von dem aus Sie via phpMyAdmin auf die Datenbank zugreifen. Wenn Sie die Voreinstellung belassen, erhalten Sie lediglich die Struktur Ihrer Datenbank angezeigt (NUR STRUKTUR). Wollen Sie den Dump als Sicherheitskopie verwenden, benötigen Sie natürlich zusätzlich die Daten (STRUKTUR UND DATEN AUSWÄHLEN). Zusätzlich können Sie noch MIT 'DROP TABLE' auswählen: Das verhindert Fehlermeldungen, wenn Sie den dump uploaden/aufspielen wollen und die gesicherten Tabellen in der Datenbank bereits vorhanden sind. Ein – gekürzter – Dump der Datenbank mysql (Mit 'drop table', Struktur und Daten) sieht z. B. so aus:

```
# phpMyAdmin MySQL-Dump
# http://phpwizard.net/phpMyAdmin/
#
# Host: localhost Datenbank : mysql
...
# -----
#
# Tabellenstruktur für Tabelle 'user'
```

```

#
DROP TABLE IF EXISTS user;
CREATE TABLE user (
    Host char(60) NOT NULL,
    User char(16) NOT NULL,
    Password char(16) NOT NULL,
    Select_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Insert_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Update_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Delete_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Create_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Drop_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Reload_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Shutdown_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Process_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    File_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Grant_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    References_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Index_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Alter_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    PRIMARY KEY (Host, User)
);
#
# Daten für Tabelle 'user'
#
INSERT INTO user VALUES( '%', 'phorumadmin', \
    '1ysr765c20qwad18', 'N', 'N', 'N', 'N', 'N', \
    'N', 'N', 'N', 'N', 'N', 'N', 'N', 'N');
INSERT INTO user VALUES( '192.103.201.%', 'dbadmin', \
    '1ysr765c20qwad18', 'Y', 'Y', 'Y', 'Y', 'Y', \
    'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');
INSERT INTO user VALUES( 'localhost', 'dbadmin', \
    '5c64a0ab236cc998', 'Y', 'Y', 'Y', 'Y', 'Y', \
    'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');
...

```

Wie sich unschwer erkennen läßt, handelt es sich bei diesem sehr transparenten Dump um eine Folge von (My)SQL-Befehlen, die gemeinsam in der Lage sind, die gesamte gesicherte Datenbank neu zu erstellen (Das Verfahren ist als Sicherungsverfahren allerdings nur für kleinere Datenbanken optimal geeignet). Nachdem ggf. eine bestehende Tabelle gelöscht worden ist (DROP TABLE), wird zunächst eine Tabelle erstellt (CREATE TABLE) und diese dann mit Werten gefüllt (INSERT). Dieses Schema wird für jede Tabelle wiederholt (im Listing nicht abgedruckt).

Noch nicht erläutert wurde der Menüpunkt QUERY BY EXAMPLE: Hierbei handelt es sich um ein neues Feature der Version 2.0.5. In dem sich nach Auswahl dieses Menüpunkts öffnenden Bildschirm können Sie sich (My)SQL-Befehle "zusammenklicken": Wenn Sie ungefähr wissen, welche Informationen Sie brauchen (und wo diese in Ihrer Datenbank zu finden sind), können Sie eine Datenbankabfrage generieren, ohne eine klare Vorstellung davon zu haben, wie Ihre Query in SQL zu formulieren wäre. Das funktioniert sogar mit JOINS (und auch mit LEFT OUTER JOINS). Ein paar Tips zum Umgang mit diesem Bildschirm:

1. QUERY BY EXAMPLE ist nur etwas für Sie, wenn Sie sich auf das Verfahren "Try and Error" einlassen können.
2. Klicken und Fummeln Sie! Keine Angst, es werden nur SELECTS abgeschickt, Ihre Datenbank bleibt also ungeschoren.
3. Nachdem Sie genug geklickt haben, müssen Sie zunächst UPDATE QUERY drücken. Die resultierende SQL-Syntax erscheint in einer Editbox (in der Sie diese auch noch verändern können).
4. Den Kreislauf "Klicken und Fummeln" mit anschließendem UPDATE QUERY beenden Sie erst, wenn Sie den Eindruck haben, daß die SQL-Syntax ein Ergebnis liefern könnte. Klicken Sie dann auf SUBMIT QUERY.

5. Sollte das zu nichts führen (es wird Ihnen lediglich der php-MyAdmin-Datenbank-Bildschirm präsentiert), dann hat Ihre Abfrage genau 0 Zeilen zurückgegeben. Verwenden Sie dann den BACK- bzw. ZURÜCK-Button Ihres Browsers und beginnen noch einmal bei 4.
6. Für mindestens ein Feld sollten Sie SHOW auswählen (sonst wird nur eine WHERE-Clausel erzeugt aber kein SELECT; was niemals eine Ausgabe erzeugen wird).
7. In das Feld Criteria schreiben Sie Bedingungen (z. B.: ='dbadmin', man beachte die Hochkommata!)
8. Die ominösen Checkboxen INS: bzw. DEL:, die sich haufenweise auf dem Bildschirm finden, beeinflussen *nicht* die Query sondern nur die Bildschirmdarstellung: Wenn Sie eine oder mehrere dieser Boxen auswählen, werden eine oder mehrere Eingabemöglichkeiten (zeilen- bzw. spaltenweise) hinzugefügt bzw. entfernt (falls Sie Ihre Abfrage weiter verkomplizieren möchten und Ihnen die bestehenden Möglichkeiten nicht ausreichen, klicken Sie einfach auf INS:). Gleiches gilt für die Auswahlboxen ADD/DELETE CRITERIA ROW bzw. ADD/DELETE FIELD COLUMNS.

Last but not Least soll der noch ausstehende Menüpunkt EIGENSCHAFTEN besprochen werden. Ein Klick auf diesen Punkt (ebenso wie übrigens der Klick auf ein Tabellename im linken Frame) bringt Sie zum phpMyAdmin-Tabellenbildschirm (vgl. Abb 9.9).

mysql009.tif/psd

Abb. 9.9: phpMyAdmin-Tabellenbildschirm.

Im phpMyAdmin-Tabellenbildschirm können Sie Tabellenspalten ändern und löschen (Klicken Sie beherzt auf die gleichnamigen Menüpunkte!). Durch Klick auf Ändern erscheint der oben bereits im Zusammenhang mit der Erstellung neuer Tabellen be-

sprochene Bildschirm (nur das Sie oben neue Werte eintragen mußten und Ihnen jetzt die bisher Gültigen als Vorgabe angezeigt werden). Wieder können Sie Primärschlüssel anlegen und Indizes erstellen (für bereits bestehende Tabellen) und diese auch wieder entfernen. Auch von hier aus können Sie sich die Inhalte aller (ANZEIGEN) oder eines Teils der Spalten (TEILW. ANZEIGEN) anzeigen lassen. Es werden zunächst die ersten 30 Zeilen ausgegeben und erst auf Ihre Anforderung hin die nächsten 30 Zeilen. Sie können dieses Verhalten übrigens verändern, indem Sie in der phpMyAdmin-Konfigurationsdatei der Variablen `$cfgMaxRows` einen anderen Wert zuweisen (z. B. `$cfgMaxRows=50;`).

Neue Felder können Sie mit NEUES FELD HINZUFÜGEN anlegen: wollen Sie drei Spalten hinzufügen, tragen Sie 3 ein und klicken auf OK. Wie im Datenbankbildschirm so können Sie auch hier ein Dump ausführen; diesmal nicht für die ganze Datenbank, sondern für die ausgewählte Tabelle. Die oben erläuterten Optionen finden Sie auch hier. Sehr praktisch sind auch die Funktionen TABELLE UMBENENNEN IN: bzw. TABELLE KOPIEREN NACH:. In Standard-SQL müssen Sie für diese eigentlich triviale, häufig anstehende Aufgabe erhebliche Verrenkungen machen, mit phpMyAdmin geben Sie einen neuen Namen an und das war es dann schon.

Eine weitere häufig benötigte Funktion ist die CSV-Ausgabe des Tabelleninhalts. Sollten Sie in die Verlegenheit kommen, eine Ihrer Datenbanktabellen in eine Tabellenkalkulation importieren zu müssen (vielleicht weil Sie einem Anwender die Möglichkeit einräumen möchten, selbst mit den Daten zu arbeiten), dann ist dies die Funktion, die Sie gesucht haben. In einer CSV-Datei (CSV steht für: *coma seperated values*) werden Ihre Datensätze (Tabellenzeilen) zeilenweise abgelegt, jeder Datensatz vom nächsten durch eine Zeilenschaltung (Newline, \n) abgetrennt. Die einzelnen Felder eines Datensatzes werden durch ein Trennzeichen getrennt (und zwar in der Regel durch ein *Semikolon* und *nicht* durch ein Komma!). Das Trennzeichen können

Sie im Übrigen in phpMyAdmin selbst bestimmen; voreingestellt ist ein Semikolon. Probleme bekommen Sie, wenn Ihre Daten selbst Semikola enthalten können.

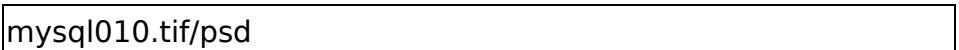
Das kommt häufiger vor, als Sie vielleicht denken, u. a. finden sich häufig HTML-Entities für die deutschen Umlaute in Datenbankfeldern (z. B. schreibt sich Müßig-Trapp in HTML so: "Müßig-Trapp". Es sei darauf hingewiesen, daß es in der Regel *keine* gute Idee ist, HTML-Entities in einer MySQL-Datenbank abzuspeichern, auch dann nicht, wenn die Datenbank als Backend für eine Website dient und mit PHP abgefragt wird. Die Vorteile, die sich hierdurch ergeben, werden in aller Regel von den Nachteilen weit übertroffen; denken Sie z. B. an Sortieralgorithmen).

Vom importierenden Programm werden die zusätzlichen Semikola als Trennzeichen erkannt und schon haben Sie Datensätze mit einer unterschiedlichen Anzahl Feldern, was üblicherweise zu *ganz* unschönen Ergebnissen führt. Enthalten Ihre Daten daher möglicherweise Semikola, müssen Sie auf ein anderes Trennzeichen ausweichen (eines, daß garantiert in Ihren Daten nicht vorkommt). Als geeignet erweist sich hier häufig die Raute (#), das Dollar- (\$) oder das Paragraphenzeichen (§). Mit den genannten Zeichen hat phpMyAdmin keine Schwierigkeiten. Dies gilt nicht für alle Zeichen: z. B. produzieren ein Stern (*) oder ein Pluszeichen (+) statt einer CSV-Datei php-Fehlermeldungen! Testen Sie also ggf. zunächst am Bildschirm, bevor Sie eine Datei mit SENDEN abspeichern.

Sollten Sie eine CSV-Datei in Microsoft Excel importieren wollen, in der ein anderes Zeichen als ein Semikolon als Trennzeichen verwendet wird (dies gilt übrigens auch für ein Komma!) dann dürfen Sie diese Datei *nicht* irgendwas.csv nennen. Tun Sie es doch, bestraft Sie Microsoft Excel, indem es alle Datenfelder eines Datensatzes als *ein* Textstring in die erste Zelle einer Zeile schreibt (was in der Regel nicht das ist, was Sie im

Sinn hatten). Speichern Sie statt dessen Ihre CSV-Datei unter dem Namen `irgendwas.txt` ab. Nur dann öffnet Microsoft Excel zunächst seinen "Text-Assistenten", dem Sie erläutern können, daß in Ihrer Datei Felder "durch Zeichen wie z. B. Kommas oder Tabulatoren" getrennt sind ("Schritt 1 von 3") und daß es sich bei diesem Trennzeichen z. B. um eine Raute handelt und MySQL, anders als manche Programme, {kein} Texterkennungszeichen verwenden, das jedes Nicht-Zahlen-Feld in Anführungszeichen einsperrt (Schritt 2 von 3"). In "Schritt 3 von 3" können Sie schließlich noch Datentypen zuweisen, falls die automatische Stringerkennung von Microsoft Excel versagt hat.

Im phpMyAdmin-Tabellenbildschirm finden Sie auch einen Menüpunkt: `TEXTDATEI IN TABELLE EINFÜGEN`. Ein Klick hierauf öffnet den in Abb. 9.10 wiedergegebenen Datenimport-Bildschirm.



mysql010.tif/psd

Abb. 9.10: phpMyAdmin-Datenimport-Bildschirm.

Dieser Bildschirm bildet eine einfach zu bedienende Schnittstelle zur leistungsfähigen MySQL `LOAD DATA LOCAL INFILE` Anweisung. Mit dieser Anweisung lassen sich auch ungewöhnlich formatierte Daten in eine MySQL-Datenbanktabelle importieren. Details zu dieser Funktion entnehmen Sie bitte dem Kapitel 9.1.1; An dieser Stelle geben wir nur einen groben Überblick. Im Textfeld geben Sie den Dateinamen an, unter dem die zu importierenden Daten abgespeichert sind (alternativ können Sie die Datei nach Klick auf `DURCHSUCHEN ...` auch in der Dateiauswahlbox auswählen). Durch Auswählen der Checkbox `TABELLENINHALT ERSETZEN`, sorgen Sie dafür, das immer dann, wenn ein Datensatz importiert wird, der in einem Feld den gleichen Inhalt hat wie ein zuvor eingelesener, und es sich bei diesem Feld um ein Primary Key oder Unique Key Feld handelt, der vorhergehende Datensatz durch den jüngsten überschrieben wird.

Wählen Sie die Checkbox nicht aus, bleibt der erste Datensatz erhalten und alle folgenden mit dem gleichen Wert in einem Primary Key oder Unique Key Feld werden verworfen. In FELDER GETRENNT MIT geben Sie das Trennzeichen an, mit dem die Spalten einer Zeile voneinander getrennt werden (meistens ein Semikolon). Sind in Ihren Daten die Felder durch Tabulatoren getrennt, geben Sie "\t" an. Unter FELDER EINGESCHLOSSEN VON können Sie angeben, durch welches Zeichen Datenfelder eingeschlossen werden (üblicherweise Anführungszeichen); wenn Sie die Checkbox OPTIONALLY auswählen, dann sind nur Textfelder von Anführungszeichen eingeschlossen. Mit FELDER ESCAPED VON können Sie ein Zeichen angeben (voreingestellt ist der Backslash \, Sie müssen ihn *zweimal* angeben, um ihn einmal einzutragen, weil er eine Sonderbedeutung hat), daß die Sonderbedeutung von Zeichen aufheben kann: Auf diese Weise (\;) könnten Sie z. B. ein Semikolon schreiben, ohne daß es als Feldtrenner interpretiert wird. Sollten Sie die Ausgabedatei auf einem anderen als auf einem Unix-System verwenden wollen, kann es notwendig sein, in ZEILEN GETRENNT MIT einen anderen Wert anzugeben.

Für DOS/Windows versuchen Sie's doch mal mit "\r\n", MacIntoshs bevorzugen "\r". Möglicherweise enthalten die Datensätze in der Datei, die Sie einlesen wollen, weniger Felder als Ihre MySQL-Tabelle. Oder die Reihenfolge der Felder ist in der Datei anders als in der Tabelle. Oder beides. In solchen Fällen hilft das Feld SPALTENNAMEN weiter: Hier geben Sie die MySQL-Spaltennamen an, in die die Daten eingelesen werden sollen (z. B. `name, vorname, email`). MySQL liest den Inhalt des ersten Feldes eines Datensatzes in die MySQL-Tabellenspalte `name`, den Inhalt des zweiten Feldes in die Spalte `vorname` und den Inhalt des dritten Feldes in die Spalte `email`. Es ist dabei völlig unerheblich, ob z. B. die Spalte `name` am Anfang, am Ende oder in der Mitte Ihrer Tabelle steht.

Die Beschreibung von phpMyAdmin ist relativ ausführlich geraten. Das war beabsichtigt. Denn phpMyAdmin funktioniert. Das

gilt nicht für alle Clients, die angeboten werden. Und phpMyAdmin wird weiterentwickelt. Auch das kann man nicht von allen Clients sagen. phpMyAdmin ist, weil es auf Clientseite nur einen Browser braucht und auf Serverseite lediglich PHP, sehr flexibel einsetzbar. Es ist geeignet, Ihre bei einem Massen-ISP gehostete MySQL-Datenbanken zu administrieren (eine andere Möglichkeit bleibt Ihnen in der Regel ohnehin nicht) Wenn Sie selbst ein ISP sind und Ihren Kunden Zugriff auf MySQL gewähren möchten, ohne das diese großen Unsinn auf dem Server anstellen können, können Sie ein zentrales phpMyAdmin installieren. Auch wenn Sie Unix-unkundigen Kollegen, Mitarbeitern oder Chefs in Ihrem Firmennetz Zugriff auf einzelne Datenbanken erlauben möchten (ggf. mit nur eingeschränkten Rechten), ohne diese mit X-Windows-Servern oder telnet-prompts zu belasten (und sich selbst mit der Installation derselben), können Sie phpMyAdmin verwenden. Wenn Sie am Wochenende notfallmäßig die Datenbank warten müssen, aber aus Sicherheitsgründen alle Ports außer Port 80 (für den WWW-Zugriff) dichtgemacht haben, können Sie mit phpMyAdmin eine Fernwartung durchführen. Die Einsatzmöglichkeiten von phpMyAdmin sind relativ breit. Aber es gibt andere grafische Clients für Linux, die ebenfalls Ihre Vorzüge haben. Wir werden dies im folgenden Abschnitt besprechen.

9.2.9 Grafische Clients unter Linux

Wir wiesen zu Beginn ja schon daraufhin: MySQL ist ein sehr populäres Datenbank-Managementsystem. Die Website <http://www.freshmeat.net> (die Suchmaschine für Linux-Software) findet für den Suchstring "mysql" mehr als 200 Einträge. Wir haben diese 200 Programme nicht alle getestet. Aber wenn Sie ein spezifisches Problem haben, stehen die Chancen nicht schlecht, daß jemand bereits eine Lösung programmiert hat und sie diese hier finden. Die meisten der Programme sind

Open Source. Sie können daher ggf. auch Programme auf Ihre Bedürfnisse anpassen. So sind beispielsweise eine Reihe von Programmen, die ursprünglich für andere Datenbanken entwickelt wurden, nachträglich für MySQL angepaßt bzw. erweitert worden (Das gilt im Übrigen auch umgekehrt: von phpMyAdmin z. B. gibt es auch eine PostgreSQL-Variante). Wir wollen Ihnen hier nur zwei der verfügbaren Clients vorstellen: den "offiziellen", von T.c.X. entwickelten Client MySQLGUI (eben weil es der offizielle grafische Client ist) und das unserer Meinung nach interessante Open Source Projekt KSql. Mit diesem für das K Desktop Enviroment (kde) entwickelte Tool können Sie nicht nur auf MySQL zugreifen sondern auch auf PostgreSQL, miniSQL und Sybase (der Sybase-Support ist allerdings noch im Alpha-Stadium).

T.c.X. MySQLGUI

MySQLGUI ist noch im Alpha-Stadium und eine ganze Reihe geplanter Features sind noch nicht implementiert. MySQLGUI ist derzeit hauptsächlich ein Query- und Reportgenerator mit eingebauten Administrationsmodul. Gegenüber phpMyAdmin hat das Programm zwei Vorteile: Es verfügt über eine eingebaute History der vom Benutzer in der Vergangenheit abgeschickten Queries und es ist in der Lage, auch große Tabellen in einem Rechenblatt auf dem Bildschirm darzustellen. Dafür fehlen dem Programm eine Vielzahl der Möglichkeiten, die phpMyAdmin bietet (so ist es beispielsweise nicht möglich, unter Nutzung der grafischen Oberfläche einzelne Datensätze zu bearbeiten).

mysql011.tif/psd

Abb. 9.11: MySQLGUI: der grafische MySQL-Client von T.c.X.

Die Installation der Binärdistribution ist denkbar einfach und beschränkt sich aufs auspacken und kopieren (die Installation der Source bringt für den Anwender keine Vorteile und wird daher hier nicht beschrieben). Sie finden das Programm auf

http://www.mysql.com/download_clients.html sowie auf der Beigleit-CD.

```
> cp /cdrom/MySQL/\
> mysqlgui-linux-static_1_6-1.tar.gz . # oder
> wget http://www.mysql.com/Downloads/mysqlgui/\
> mysqlgui-linux-static.1.6-1.tar.gz
```


Auspacken (`gunzip`), ent"tar"en (`tar xvf`) und das im Verzeichnis enthaltene Binary (`mysqlgui`) an passende Stelle kopieren (z. B. nach `/usr/local/bin`) – fertig. Wenn Sie wollen, können Sie auch die Dokumentation (`MySQL.help`) an eine öffentlich zugängliche Stelle kopieren (z. B. `/usr/local/lib/mysqlgui/`). Dann kann jede und jeder auf diese Datei zugreifen, jedenfalls wenn die Konfigurationsdateien entsprechend angepaßt werden. Womit wir schon beim nächsten Thema wären: In den bereits im Kapitel XX diskutierten MySQL-Konfigurationsdateien können Sie auch das Verhalten von MySQLGUI beeinflussen. Erstellen Sie zunächst einen `[client_ftk]`-Eintrag in der Datei `/etc/my.cnf` (oder wo immer in Ihrer Distribution die systemweit gültigen Konfigurationsdateien abgelegt werden). In diesem können Sie dann die folgenden Eintragungen vornehmen:

```
[client_ftk]
help_file=/usr/local/lib/mysqlgui/MySQL.help
client_file=~/.MySQLGUI.options
history_length=20
database=mysql
queries_root=~/.MySQLGUI.queries
last_database_file=~/.MySQLGUI.ladb
```

Die Beispiel-Konfigurationsdatei, die mit MySQLGUI kommt, schlägt als Standardverzeichnis für die benutzerspezifischen Dateien `~/.guiclient` vor (die Tilde steht wie üblich für das Heimatverzeichnis des Benutzers). Dies ist aber erstens kein sehr sprechender Name und zweitens muß dieses Verzeichnis zunächst von jedem Benutzer angelegt werden, bevor es be-

nutzt werden kann (MySQLGUI legt ein fehlendes Verzeichnis *nicht* an). Aus diesem Grund schlagen wir die oben wiedergegebene, veränderte Konfiguration vor. MySQLGUI reagiert mit Absturz, wenn Sie z. B. den Eintrag `database=xyz` entfernen. Geben Sie also immer eine Default-Datenbank an. Insgesamt reagiert MySQLGUI etwas empfindlich auf Veränderungen im Konfigurationsblock. Es empfiehlt sich daher, Änderungen zunächst zu testen. Die History wird Standardmäßig in der Datei `~/ .mysql_history` abgelegt (Sie können das durch den Eintrag `set-variable = MYSQL_HISTFILE=~/ .wunschname` in der Datei `my.cnf` ändern). Diese Datei wird auch vom Standard-Client `mysql` verwendet. Queries, die Sie in einem der beiden Clients verwendet haben, stehen daher im jeweils anderen auch zur Verfügung.

Nach dem ersten Start von MySQLGUI sollten Sie zunächst unter Optionen den Reiter `CLIENT` auswählen, Ihren MySQL-Benutzernamen eintragen sowie die Option `ASK FOR PASSWORD` aktivieren. Nach Klick auf `SAFE` verlassen Sie das Programm am besten wieder, nur um es erneut zu starten. Nun erst werden Sie (wie bei allen folgenden Programmstarts) nach Ihrem Kennwort gefragt (was vielleicht lästig ist, aber die Voraussetzung für den Zugriff auf Ihre Datenbanken).



mysql012.tif/psd

Abb. 9.12 MySQLGUI-Administrator-Panel.

Im oberen Teil des Edit-Fensters des MySQLGUI-Hauptbildschirms formulieren Sie Ihre Datenbankabfragen. Bei Bedarf können Sie sich aus dem unteren Teil (der History, Ihrem "Abfragegedächtnis") schon einmal formulierte Abfragen in den oberen Teil holen. Der Klick auf die mittlere Maustaste öffnet ein Popup-Menü, aus dem Sie alle MySQL-Befehle, Funktionen usw. auswählen können. Wenn Sie zuvor eine Tabelle ausgewählt haben (`COMMANDS` | `TABLES` | `CHOOSE TABLE`) erscheint nach Klick auf die rechte Maustaste ein Menü, aus dem Sie alle Spalten der Tabelle auswählen können. Das Administrator-Panel

(Abb. 9.12) vereinigt alle administrativen Funktionen auf einem Bildschirm; ein Klick auf `COMMANDS | ADMINISTRATION | ADMINISTRATORS PANEL` fördert es zu Tage (sofern Sie sich als MySQL-Benutzer mit administrativen Rechten eingeloggt haben). Von hier aus können Sie die wichtigsten administrativen Aufgaben ausführen: Es ist möglich, Datenbanken zu erstellen und zu löschen, sie neu zu laden und herunterzufahren usw. Auch eine Überwachung der wichtigsten Datenbankparameter ist möglich (Datenbankstatus anzeigen, Prozessliste anzeigen). Ein Modul zur deutlich vereinfachten Benutzerverwaltung sowie eine Funktion zum schnellen Setzen neuer Kennworte steht ebenfalls zur Verfügung.

mysql013.tif/psd

Abb. 9.13 MySQLGUI Datenbankabfrage-"Rechenblatt".

Die "Rechenblätter", in denen Datenbankabfragen dargestellt werden haben eine Reihe nützlicher Eigenschaften: Sie sind in der Lage, auch sehr große Tabellen aufzunehmen und diese in eine Datei zu exportieren. Die Größe der Tabellen ist nur begrenzt vom virtuellen Arbeitsspeicher Ihres Rechners. Eine Tabelle mit 248.000 Einträgen wurde auf unserem Referenzsystem (Pentium III, immerhin nach 28 Sekunden angezeigt und ließ sich problemlos durchscrollen (allerdings nicht sortieren). Auch das Sichern des Tabelleninhalts in eine gut 80 Mbyte große Datei erwies sich als problemlos und dauerte etwas mehr als 30 Sekunden. Das Exportformat können Sie übrigens im Optionenbildschirm beeinflussen (Reiter `FILE`). Weitere, etwas versteckte Funktionen: Ein Klick auf die Spaltennamen sortiert die Tabelle nach dieser Spalte, der erste Klick aufsteigend, der zweite absteigend. Klicken sie mit der mittleren Maustaste auf einen Spaltennamen, öffnet sich ein Suchen-Dialog, mit dessen Hilfe Sie die betreffende Spalte durchsuchen können.

KSql (früher KMySQL)

KSql ist ein Datenbankclient für KDE. Ursprünglich hatten die Entwickler lediglich einen Client für MySQL im Sinn (daher KMySQL). Inzwischen haben sie ihren Anspruch erweitert und stellen mit KSql einen grafischen Datenbankclient für eine ganze Reihe von Datenbanken zur Verfügung (miniSQL, PostgreSQL, Sybase), wobei die Unterstützung für MySQL am fortgeschrittensten ist. Die Integration verschiedener Datenbanken unter einer Oberfläche kann dabei als ausgesprochen gelungen gelten: In einer Baumstruktur werden gleichzeitig alle verfügbaren Datenbanken unter einem Dach angeboten (dabei können z. B. MySQL- und PostgreSQL-Datenbanken gemischt werden). Auf diese Weise ist der Zugriff auf verschiedene Datenbanken mit einheitlichen Werkzeugen möglich. Bestechend ist auch die Möglichkeit, mit Hilfe eines grafischen Editors Formulare zu erstellen, mit denen auch SQL-unkundige Benutzer (von einem Datenbankadministrator vorher erstellte) komplexe Datenbankabfragen verwenden können. Dabei kann auf Pulldown-Menüs, Ankreuz- und Editierfelder und sogar auf einen Kalender zurückgegriffen werden. Als besonderes Highlight wird inzwischen auch eine php-Bibliothek zur Verfügung gestellt (KMySQL-PHP), mit der "ultra-easy" der Zugriff auf KMySQL-Formulare über einen Web-Browser ermöglicht werden kann.

Die Installation des Client-Kerns aus der Source-Distribution bereitet keine Schwierigkeiten und besteht (nach dem Kopieren und Entpacken) aus dem üblichen Dreiklang:


```
> cp /cdrom/kSQL/kmysql-1_2_1_tar.gz . # oder
> wget http://ksql.sourceforge.net/download/\
> kmysql-1.2.1.tar.gz
> gunzip kmysql-1.2.1.tar.gz
> tar xvf kmysql-1.2.1.tar
> cd kmysql-1.2.1
> ./configure
```

```
> make
> make install
```

Schwierigkeiten bereitet hingegen die Installation des MySQL-Moduls (KSql verfolgt ein modulares Konzept: nach der Grundinstallation des Programmkerns muß mindestens ein Datenbankmodul installiert werden. Zur Auswahl stehen Module für MySQL, miniSQL, postreSQL sowie, noch im Alpha-Stadium, für Sybase). Die Installation des RPM-Archivs gestaltet sich problemlos (auf der CD finden Sie jeweils eine RPM-Datei für Red-Hat-basierte Distributionen sowie für SuSE).

```
> rpm --install /cdrom/kSQL/kmp_mysql-1_2_0-1_i386.rpm
> # bzw. für SuSE:
> rpm --install /cdrom/kSQL/kmp_mysql-1_2_0-1_i386_SuSE.rpm
```

Nach der Installation können Sie das Programm mit `kmysql` aufrufen (*nicht* mit `ksql`, dies ist wohl noch ein Relikt aus den Nur-MySQL-Zeiten von KSql). Wenn Sie unter `OPTIONS | SETTINGS` die Checkbox "Save Settings before quitting" auswählen, brauchen Sie nicht bei jedem Aufruf erneut das Datenbankmodul und die zu verwendende Datenbank auszuwählen. Allerdings sollten Sie dann in der nach Klick auf `SERVER | ADD SERVER` erscheinenden Dialogbox keinesfalls ein Paßwort angeben: dieses würde nämlich im Klartext in der Konfigurationsdatei von KSql gespeichert. Ein eventuell bereits eingetragenes Paßwort werden Sie nur durch Editieren der Konfigurationsdatei "per Hand" wieder los. Löschen Sie ggf. in der Datei `~/.kde/share/config/kmysqlrc` ihr Paßwort wieder (nach Ihrem MySQL-Benutzernamen folgen dann einfach zwei Kommata).



mysql014.tif/psd

Abb. 9.14 KSql (früher KmySql).

Die Benutzeroberfläche von KSql entspricht dem KDE-Standard; wer auf sein Linux-System regelmäßig über den KDE-Desktop zugreift, wird sich auf Anhieb zurechtfinden. Unterhalb der obli-

gatorischen Menü- und der ebenso obligatorischen Iconleiste finden sich die vier Hauptfenster von KSql: Auf der linken Seite das Datenbankfenster (1), das in vertrauter Manier sämtliche DBM-Systeme, auf die zugegriffen werden kann, in einer Baumstruktur anzeigt. Je nach Ebene öffnen sich Datenbanken, Tabellen oder Spalten. Ein Doppelklick auf ein Tabellensymbol führt ein `SELECT * from tabellename` aus (es wird der Inhalt der Tabelle im Ausgabefenster angezeigt). Ein Doppelklick auf ein Spaltensymbol gibt Eigenschaften einer Spalte aus (einschließlich der Rechte, mit denen der jeweilige Benutzer auf diese Spalten zugreifen darf). Ein Klick mit der rechten Maustaste öffnet kontextsensitive Popup-Menüs. Haben Sie ein Datenbankicon ausgewählt, wird Ihnen beispielsweise die Möglichkeit angeboten, eine neue Tabelle anzulegen oder mit Hilfe des `SELECT WIZARDS` eine Datenbankabfrage zu formulieren. Der Rechtsklick auf ein Tabellensymbol fördert ein Popup-Menü zu Tage, das unter anderem den wichtigen Eintrag `EDIT CONTENTS` verbirgt. Ein Klick hierauf öffnet die Tabelle in einem editierbaren Spreadsheet. Auf diese Weise sind Änderungen an einzelnen Datensätzen in bestehenden Tabellen schnell und einfach und ohne `UPDATE`-Befehl möglich. Ein Rechtsklick auf ein Spaltenicon öffnet schließlich ein Popup-Menü, aus dem heraus sich mit Hilfe der Option `Edit column` der Spalteneditor auswählen läßt (mit dem Sie alle Spalteneigenschaften verändern können). In allen Rechtsklick-Popup-Menüs finden sich außerdem Funktionen zum Erstellen oder zum Löschen der jeweiligen Objekte (Datenbanken, Tabellen, Spalten).

Das größte Fenster (2) nimmt die jeweiligen Ausgaben auf, unter anderem auch die Ausgaben beliebiger Queries. Ausgabefenster können Sie sich beliebig viele anlegen (durch Angabe eines Namens und `RETURN` im kleinen Editfeld links oben) und durch Klick auf die entsprechenden Reiter (3) nach vorne holen. Den Inhalt der Ausgabefenster (z. B. die Inhalte Ihrer Tabellen) können Sie durch Auswahl der Option `View | Html export` auch

in eine HTML-Datei exportieren. Für die schnelle Quick-and-Dirty-Tabelle ("... können Sie mal eben ...") ist das oftmals völlig ausreichend bzw. eine gute Vorlage.

Im Eingabefenster (4) können Sie nach Belieben Datenbankabfragen formulieren. Bereits abgeschickte Queries können Sie aus dem History-Fenster (5) heraus erneut ausführen bzw. in das Eingabefenster befördern, um sie zu bearbeiten. Schauen Sie sich auch mal die Editor-Optionen an (OPTIONS | EDITOR OPTIONS ...).

Ein Rechtsklick auf eines der DBMS-Symbole (6) ermöglicht Ihnen auch den Zugriff auf einige administrative Funktionen: Sie können den Datenbankserver herunterfahren, neu starten und sich die wichtigsten Informationen zum Serverstatus anzeigen lassen. Eine der interessantesten Features von KSql ist aber sicher die Möglichkeit, eigene Statische Abfragen (8) bzw. Formulare (9) zu generieren und abzuspeichern. Da sämtliche der zum Aufbau der Formulare erforderlichen Informationen in Hilfstabellen in der Datenbank selbst abgespeichert werden, ist das Konzept prinzipiell universeller als eine reine clientseitige Lösung, weil nicht konkrete Abfragen im Client implementiert werden müssen sondern allgemeingültige Funktionen, die aus den Informationen in den Hilfstabellen ein konkretes Formular generieren. Auf diesem Weg erstellt auch die weiter unten besprochene KMySQL-PHP-Bibliothek Formulare für den WebBrowser.

mysql015.tif/psd

Abb. 9.15 KSql-Formular-Editor.

Der KSql-Formular-Editor wird durch Rechtsklick auf die Datenbank, für die Sie ein Formular erstellen wollen und Auswahl des Menüpunkts NEW FORM aufgerufen. Es öffnet sich Ihnen das in Abb. 9.15 gezeigte Editorfenster. Die gerasterte Grundfläche können Sie nach Belieben und Bedarf vergrößern bzw. verklei-

uern. Sie stellt die Grundfläche für Ihr neues Formular dar. Durch Klick auf eines der sechs Symbole am linken Rahmenrand befördern Sie ein entsprechendes Objekt auf die Formulargrundfläche. Jedem Objekt, daß Sie in Ihrem Formular verwenden, müssen Sie einen Namen geben. Mit Hilfe dieses Namens können Sie in der dem Formular hinterlegten Datenbankabfrage die vom Benutzer gesetzten Werte abfragen. Die Objekte können die folgenden Eigenschaften erhalten (die Sie ggf. durch entsprechende Angaben in einem nach Rechtsklick auf das Objekt sich zeigenden Formular zuweisen müssen):

- *Buttons*: Ein Button kann drei Arten von Ereignissen in Ihrem Formular auslösen: *Submit* schickt eine Datenbankabfrage ab ("Anzeigen" in Abb. 9.15), *Cancel* schließt das Formular ohne weitere Aktivität und *Help* zeigt den Hilfetext, den Sie in den Voreinstellungen des Formulars angegeben haben (sofern Sie einen angegeben haben, siehe unten).
- *Labels*: Ein Label enthält lediglich (erläuternden) Text (Wie z. B. "Bitte wählen Sie eine Rubrik aus!" in Abb. 9.15)
- *Text fields*: Mit Hilfe dieses Objekts können Sie vom Benutzer Textangaben anfordern (wie im Edit-Feld unter "oder suche nach ..." in Abb. 9.15.)
- *Combo boxes*: Hierbei handelt es sich um die von vielen Entwicklungsumgebungen her bekannten Auswahllisten. Entweder Sie geben alle Werte "händisch" an; dann wird später Ihrem Skript bei der Datenbankabfrage der ausgewählte Textstring als Wert übergeben. Sie können aber auch eine SQL-Query angeben, die die gewünschten Werte aus einer Datenbank holt. In diesem Fall kann die Datenbankabfrage einen aber auch zwei Spalten zurückgeben. Wird eine Spalte zurückgegeben, entspricht der Wert des Objekts dem ausgewählten Text. Werden jedoch zwei Spalten zurückgegeben, verwendet KSql den Inhalt der

ersten Spalte als den in der Combo box anzuzeigenden Text und den Inhalt der zweiten Spalte als den bei der späteren Abfrage zurückzugebenden Wert. Das ist z. B. dann vorteilhaft, wenn Sie in einer Tabelle einen Schlüssel verwenden, der in einer anderen (Schlüssel)-Tabelle aufgelöst wird (vgl. Sie hierzu das Beispiel unten).

- *Check boxes*: Dies sind Ankreuzfelder bzw. Auswahlfelder. Im Einstellungs Menü jeder Check box können Sie wählen, ob die Boxen bei der Darstellung in Ihrem Formular bereits angekreuzt (angehakt) sein sollen oder nicht (INITIAL-STATE). Für die beiden möglichen Zustände einer Check box müssen Sie jeweils die Werte angeben (unter CHECKED VALUES den Wert, der übergeben wird, wenn die Box ausgewählt wird und unter UNCHECKED VALUES den Wert für den Fall, daß die Box nicht ausgewählt wird).
- *Date pickers*: Aus diesem Kalenderobjekt können die Benutzer des Formulars ein Datum auswählen. Als Wert wird das Datum im MySQL Format übergeben (YYYY-MM-DD).

Ein Beispiel:

Sie haben eine Datenbank Software mit zwei Tabellen. Die erste Tabelle (soft) enthält Spalten für den Namen eines Programms, für einen Code für die Kategorie, zu der das Programm gehört (z. B. Grafik, Textverarbeitung etc.), für eine ausführliche Beschreibung des Programms usw. Die zweite (Schlüssel-) Tabelle löst die Codes für die Kategorien in Klartext auf.

```
# Tabellenstruktur für Tabelle 'soft'
CREATE TABLE soft (
  id int(10) NOT NULL auto_increment, # Ident-Nr.
  name varchar(120) NOT NULL,        # Programmname
  rubrik int(10) NOT NULL,           # Rubrik
  beschreibung text NOT NULL,        # Beschreibung
```

```

    url varchar(120) NOT NULL,           # Lokale Adresse
    homepage varchar(120),              # Homepage
    zeit varchar(14),                   # Eintragsdatum
    PRIMARY KEY (id)
);

# Tabellenstruktur für Tabelle 'rubrik'
CREATE TABLE rubrik (
    schluessel int(10) NOT NULL auto_increment, # Code
    name varchar(120) NOT NULL,                # Klartext
    PRIMARY KEY (schluessel)
);

```

Die Spalte `rubrik` der Tabelle `soft` ist ein Fremdschlüssel (auf den Primärschlüssel der Tabelle `rubrik`). Einen vollständigen SQL-Dump der Datenbank `Software` (Struktur und Daten) finden Sie übrigens auf der Buch-CD (Verzeichnis `Beispiel-DBs/Software.sql`); da sämtliche Informationen der KSql-Formulare in zwei Hilfstabellen dieser Datenbank abgespeichert werden, erhalten Sie damit gleichzeitig auch das KSql-Formular. Angenommen, Sie möchten ein Formular wie das in Abbildung 9.16 erstellen, mit dessen Hilfe ein Anwender sich z. B. alle Programme einer Rubrik anzeigen lassen kann oder alle Programme, in deren Beschreibung oder Titel ein Suchstring vorkommt. Außerdem soll der Anwender selbst bestimmen können, ob er z. B. die Programmbeschreibung ausgegeben bekommen möchte oder nicht. Dann rufen Sie zunächst den Formuareditor auf (Rechtsklick auf das Datenbanksymbol der Datenbank `Software`, `NEW FORM` auswählen), klicken sich die erforderlichen Objekte zusammen und verteilen diese malerisch auf Ihre Arbeitsfläche.

In den Menüs, die sich nach Rechtsklick auf die Objekte und Auswahl von PROPERTIES zeigen, machen Sie jeweils die erforderlichen Angaben. In den Einstellungen der Combo box wählen Sie GET CHOICES FROM QUERY: und geben die folgende Datenbankabfrage ein:

```
SELECT name, schluessel FROM rubrik ORDER BY name
```

Dies füllt die Combo box mit den entsprechenden Einträgen. Als nächstes müssen Sie Ihre eigentliche Datenbankabfrage formulieren. Dabei können Sie die Benutzereingaben in Ihrer Query mit der Syntax \$objektname\$ ansprechen:

```
SELECT soft.name 'Programm'
       $Download$
       $Homepage$
       $Beschreibung$
FROM soft
WHERE soft.rubrik = '$Rubrik$' OR
      ( (soft.name like '%$Suchstring$%' OR
        soft.beschreibung like '%$Suchstring$%') AND
        '$Suchstring$' != '' )
ORDER BY soft.name
```

Damit das so funktioniert, müssen Sie natürlich die abgefragten Objekte entsprechend benannt haben: \$Download\$, \$Homepage\$ und \$Beschreibung\$ sind die Namen der drei Checkboxen unseres Formulars. Ihre Werte sind (für den "Checked"-Fall; im "Unchecked"-Fall werden leere Strings übergeben):

```
, soft.url "Lokale Download-Adresse" # $Download$
, soft.homepage "Homepage"           # $Homepage$
, soft.beschreibung "Beschreibung"   # $Beschreibung$
```

Die führenden Kommata sind erforderlich! Nach dem ersten OR in der WHERE-Klausel folgt die Abfrage des Suchstrings (die Abfrage wird nur relevant, wenn der Suchstring nicht leer ist). Der Programmname (`soft.name`) wird in jedem Fall ausgegeben, alle anderen Felder nur, wenn die entsprechende Check box ausgewählt wurde (wenn nicht, wird einfach ein leerer String eingetragen. Das stört unsere Query nicht). Wenn Sie wollen, können Sie abschließend in den Formular-Properties noch einen Hilfetext angeben (damit der Anwender nach Klick auf HILFE auch wirklich geholfen wird). Wenn Sie den SQL-Dump Software.sql von der CD einspielen, steht Ihnen das Formular auch dann zur Verfügung, wenn Sie keine Lust hatten, das Beispiel im Detail nachzuvollziehen.

Abschließend soll die KMySql-PHP Bibliothek vorgestellt werden. Mit ihr ist es möglich, KSql-Formulare (wie das in Abb. 9.16) ohne großen Aufwand auch im eigenen Web-Angebot zur Verfügung zu stellen. Auf dem System, auf dem die Webseite installiert werden soll, muß PHP sowie das MySql-Modul für PHP zur Verfügung stehen. Im Kapitel 9.1.4 (phpMyAdmin) wurde erläutert, wie geprüft werden kann, ob diese Voraussetzungen gegeben sind (HTML-Seite mit `<?php phpinfo() ?>`). Es werden an dieser Stelle keine grundlegenden Erklärungen zu PHP gegeben. Genauere Informationen zu PHP finden Sie im Kapitel 13.2.

Die Installation von KMySql-PHP ist denkbar einfach: Es ist lediglich das Kopieren der Datei `kmysql.php3` in ein Verzeichnis erforderlich, auf das der Apache-Web-Server zugreifen kann. Es ist gute Sitte, Bibliotheksdateien nicht im `Document Root` des Web-Servers vorzuhalten. Angenommen, `Document Root` ist `/usr/local/www/htdocs`, dann bietet sich für Bibliotheksdateien ein Verzeichnis `/usr/local/www/php` an. Wenn Sie dieses Verzeichnis in Ihrer PHP-Konfigurationsdatei `php3.ini` bzw. `php.ini` (üblicherweise in `/usr/local/lib`) als `Include Path` festlegen (tragen Sie z. B. folgendes ein: `include_path =`

./usr/local/www/php), dann findet PHP die Bibliothek auch dann, wenn ihr Aufruf ohne absolute Pfadangabe erfolgt.

```
> cp /cdrom/kSQL/KMySQL-PHP-0_1_0_tar.gz . # oder
> wget http://ksql.sourceforge.net/download/\
>   KMySQL-PHP-0.1.0.tar.gz
> gunzip < KMySQL-PHP-0.1.0.tar.gz | tar xvf -
> cd cd KMySQL-PHP-0.1.0
> cp kmysql.php3 /usr/local/www/php
```

Die Bibliothek stellt drei Funktionen zur Verfügung, mit deren Hilfe auf KSql-Formulare zugegriffen werden kann:

ksql_create_form(string host, string user, string password, string base, string form_name, string action)		
Parameter	Beispiel	Erläuterung
host	"localhost"	Name des Datenbank-Servers
user	"everyone"	MySQL-Benutzername, mit dem auf die Datenbank zugegriffen werden kann
password	"egal"	Das zum Benutzernamen passende Kennwort
base	"software"	Name der Datenbank, daß das Formular enthält, das dargestellt werden soll.
form_name	"Software nach Rubrik"	Name des Formulars, das dargestellt werden soll.
action	"ergebnis.php3"	Der URL, der aufgerufen werden soll, wenn das Formular abgeschickt werden soll.

Die Funktion `ksql_create_form` erstellt ein komplettes HTML-Formular entsprechend einem zuvor definierten KSql-Formular. Die Funktion ist in eine HTML-Seite einzubetten. Ergeben sich beim Funktionsaufruf Probleme, wird eine Fehlermeldung ausgegeben.

<code>ksql_run_form(string host, string user, string password, string base, array post_vars)</code>		
Parameter	Beispiel	Erläuterung
<code>host, user, password, base</code>	s. o.	s. O.
<code>post_vars</code>	<code>\$HTTP_POST_VARS</code>	Array der Variablen, die mit dem Server mittels POST übergeben werden. In der Regel werden Sie das PHP-Array <code>\$HTTP_POST_VARS</code> verwenden. Damit das funktioniert, muß <code>php_track_vars</code> (z. B. in der <code>php.ini</code>) gesetzt sein.

Diese Funktion fügt die mit dem Formular erfragten Benutzerangaben in die SQL-Query ein und übergibt diese dem Server. Die Funktion gibt ein assoziatives Array zurück. Die Rückgabewerte brauchen Sie normalerweise nicht selbst zu behandeln (Details können Sie ggf. der Dokumentation entnehmen). Das Rückgabearray dieser Funktion wird im Regelfall einfach der Funktion `ksql_print_result` übergeben.

<code>ksql_print_result(array result)</code>		
Parameter	Beispiel	Erläuterung
<code>result</code>	<code>\$result</code>	Dies ist der Rückgabewert des Aufrufs

		der Funktion \$result = ksql_run_form(...)
--	--	---

ksql_print_result gibt im Mißerfolgsfall eine Fehlermeldung aus und ruft im Erfolgsfall eine Funktion ksql_print_result_to_table auf, die ihrerseits eine HTML-Tabelle mit dem Inhalt der Datenbankabfrage erzeugt. Um Details brauchen Sie sich dabei nicht kümmern.

Da es nicht sinnvoll ist, Kennworte in öffentlich zugänglichen Verzeichnissen abzuspeichern, sollte die Account-Informationen, die die besprochenen KMySQL-PHP-Funktionen benötigen, nicht direkt in der HTML-Datei abgelegt werden. Besser ist es, diese Informationen in einer gesonderten Datei abzulegen, die nicht in Document Root gespeichert wird, sondern – wie die Bibliothek selbst – in einem PHP Include Directory. Eine Datei mit dem folgenden Inhalt sollte daher z. B. nach /usr/local/www/php/kmysql-php_setup.inc kopiert werden:

```
<?php
# Datei kmysql-php_setup.inc
# Hier werden die Zugriffsdaten hinterlegt
$host="localhost";
$user="everyone";
$password="egal";
?>
```

Die Angaben müssen natürlich auf die bei Ihnen vorliegenden Verhältnisse angepasst werden. Was jetzt noch gebraucht wird, ist eine öffentlich zugängliche PHP-Datei, die das Formular ausgibt, die SQL-Query abschickt und ggf. die von der Datenbank zurückgegebenen Daten in einer HTML-Tabelle darstellt. Eine solche Datei könnte z. B. folgendermaßen aussehen (Sie finden diese Datei kmysql-php_formular.php³ auch auf der CD im Verzeichnis Beispiel-Skripts):

```

<html>
<head>
  <title>Software Recherche</title>
</head>
<body>
<h2>Software Recherche</h2>

<?php
# laden der KMySql-PHP-Bibliothek und der
# Datei mit den Zugriffsdaten
require("kmysql.php3");
require("kmysql-php_setup.inc");

# Funktion erstellt das Formular, die Variable
# PHP_SELF enthält den Namen der aufgerufenen Datei,
# die Datei ruft sich also selber auf
ksql_create_form($host, $user, $password,
                 "software", "Software nach Rubriken",
                 $PHP_SELF);

# Diese Funktionen sollen nur aufgerufen werden, wenn
# der Submit-Button gedrückt wurde (deswegen die
# if-Abfrage). die ksql-Funktionen sorgen für die
# Formulierung der SQL-Abfrage und die Darstellung der
# Datenbankausgabe in einer HTML-Tabelle.
if ($HTTP_POST_VARS){
  $result=ksql_run_form($host, $user, $password,
                       "software", $HTTP_POST_VARS);
  ksql_print_result($result);
}
?>

</body>
</html>

```

Durch die Kommentare wirkt das Skript gewaltiger als es ist. Im Grunde sind tatsächlich nur drei Funktionsaufrufe erforderlich, die dann für eine Ausgabe sorgen wie in Abb. 9.17 gezeigt. Die Kommentare im Skript bedürfen keiner weiteren Erklärung.

```
mysql017.tif/psd
```

Abb. 9.17: KMySQL-PHP ermöglicht die einfache Verwendung von KSql-Formularen auch in Web-Browsern.

9.2.10 Microsoft-Windows-Programme als MySQL-Client

T.c.X. stellt mit MyODBC einen Treiber für Windows 95 und Windows NT zur Verfügung, mit dem es problemlos möglich ist, von Windows aus auf eine MySQL-Datenbank zuzugreifen. Dabei spielt es keine Rolle, wo und unter welchem Betriebssystem der Datenbankserver läuft. Dies kann überall in Ihrem Firmennetz, im Intranet oder im Internet sein. Sie können also problemlos den MySQL-Datenbankserver auf einer Linux-Workstation installieren und andere Benutzer können dennoch über Ihre Windows-Heimcomputer auf die Datenbanken zugreifen.

Zunächst muß auf jedem Windows-Gerät der MyODBC-Treiber installiert werden. Für Windows 95 und Windows NT gibt es verschiedene Binaries, die Sie im Web auf der Homepage von T.c.X (http://www.mysql.com/download_myodbc.html) sowie auf CD finden. Für die Installation unter Windows NT müssen Sie sich als Administrator einloggen. Nach dem Entpacken des ZIP-Archives muß das obligatorische `setup.exe` gestartet werden. Nach ein paar sich selbst erklärenden Klicks ist die Installation bereits abgeschlossen. Sie müssen den Treiber nun zunächst für den Zugriff auf Ihre Datenbank(en) konfigurieren. Klicken Sie dazu auf **START | EINSTELLUNGEN | SYSTEMSTEUERUNG** und Doppelklicken in dem sich öffnenden Fenster auf **ODBC-DATENQUELLEN**. Es öffnet sich der "ODBC-Datenquellen-Administrator". Wählen Sie den Reiter **BENUTZER-DSN** aus. Sollte sich im Fenster bereits ein Eintrag `sam-`

ple-MySQL (Treiber: MySQL) befinden, wählen Sie diesen aus und klicken auf KONFIGURIEREN. Gibt es noch keinen Beispiel-Eintrag, müssen Sie zunächst auf HINZUFÜGEN klicken; im sich öffnenden Fenster "Neue Datenquelle erstellen" wählen Sie MySQL aus und klicken auf FERTIGSTELLEN. Es öffnet sich Ihnen ein Fenster mit dem Titel TDX mysql Driver Default Configuration. In diesem tragen Sie für Windows DSN name einen beliebigen Namen ein (z. B. Software-MySQL, wenn auf die Beispieldatenbank "Software" zugegriffen werden soll). Unter MySQL host (name or IP) geben Sie den Hostnamen oder die IP-Adresse des Datenbank-Server-Hosts an (z. B. datenbank.ihre-domain.de). Im Feld MySQL database name geben Sie den Namen der MySql-Datenbank an, auf die Sie zugreifen wollen (z. B. Software). Schließlich müssen Sie noch den Benutzernamen angeben, unter dem Sie sich einloggen wollen (z. B. User: everyone). Denken Sie daran, diesem Benutzer in der Rechteverwaltung von MySql den Zugriff auf die Datenbank vom Host des Clients aus zu gestatten. Schicken Sie ggf. die folgende Query ab:

```
mysql> INSERT INTO db VALUES('%', 'software', 'everyone',  
    'Y', 'Y', 'Y', 'Y', 'N', 'N', 'N', 'N', 'N', 'N');  
mysql> FLUSH PRIVILEGES; # nicht vergessen!
```

Tragen Sie *kein* Kennwort ein. Abgespeicherte Kennwörter sind fast immer eine schlechte Idee. Sie haben später beim Connect die Möglichkeit, das Paßwort anzugeben. Wenn Sie eine ältere MySql-Server Version installiert haben, müssen Sie als Port 3333 eintragen. Für alle MySql-Versionen ab 3.21 tragen Sie 3306 ein (sofern Sie den Server nicht abweichend konfiguriert haben). Damit ist die ODBC-Konfiguration abgeschlossen.

In Microsoft Access (Access 97, für andere Versionen ist möglicherweise eine etwas andere Vorgehensweise erforderlich) erstellt man sich nun zunächst eine leere Datenbank (Der "Assistent" bietet diese Möglichkeit an). Das Folgende ist nicht eigentlich schwierig, die Optionen sind nur jeweils (wie häufig in

Microsoft-Programmen) gut versteckt. Unter Windows ist Geduld bekanntlich eine Kardinaltugend.

Klicken Sie auf DATEI | EXTERNE DATEN | TABELLEN VERKNÜPFEN In der sich öffnenden Dateiauswahlbox klicken Sie in der Combo-Box neben Dateityp auf ODBC-DATENBANKEN(). Daraufhin verschwindet die Dateiauswahlbox wieder, stattdessen erscheint ein Fenster namens `Datenquelle auswählen`. Klicken Sie auf den Reiter `COMPUTERDATENQUELLE`. In dem sich nun öffnenden Fenster finden Sie Ihre zuvor erstellte Datenquelle `Software-MySQL` wieder! Klicken Sie doppelt drauf. Nun öffnet sich das Konfigurationsfenster der Datenquelle, indem Sie das Paßwort für den Zugriff eintragen können (und müssen). Nach Klick auf OK öffnet sich ein Fenster `Tabellen verknüpfen`. Klicken Sie `AUF ALLE AUSWÄHLEN` und anschließend auf OK.

Sie können nun die Tabellen doppelklicken, die Ihnen daraufhin in Spreadsheets gezeigt werden. Sie haben vollen (lesenden wie schreibenden!) Zugriff auf die Datenbank, d. h. Sie können auch Änderungen vornehmen (sofern Sie serverseitig über die entsprechenden Rechte verfügen). Unter Verwendung von Microsoft Access können Sie all das mit Ihrer MySQL-Datenbank veranstalten, was die Werkzeuge von Microsoft Access ermöglichen. Wir haben z. B. einen Assistenten für die Erstellung von Berichten bemerkt (Berichts-Assistent). Dieser erscheint nach Klick auf `BERICHTE | NEU | BERICHTS-ASSISTENT`. Vielleicht lassen sich mit diesem Berichte erstellen? Konsultieren Sie im Zweifelsfall die von Microsoft mitgelieferte Dokumentation oder den zappeligen Büroklammerassistenten oder den Windows-Systemadministrator.

Beim nächsten Mal, wenn Sie den Wunsch verspüren, via Microsoft Access auf MySQL-Datenbank `Software` zuzugreifen, brauchen Sie diese Odysee nicht noch einmal durchmachen: Doppelklicken Sie einfach auf die Microsoft Access Datenbank `Software.mdb`. (oder unter welchem Namen auch immer Sie die Mi-

rosoft Access-Datenbank abgespeichert haben). Nach Eingabe Ihres Kennwortes haben Sie Zugriff und auch der ggf. durch den Berichts-Assistenten erstellte Bericht ist noch vorhanden.

Wenn Sie mit Microsoft Access arbeiten, dann sollten Sie sich auch das wirklich nützliche Freeware-Tool MyAccess von Hubertus Hiden gönnen. Sie finden es auf der MySQL-Homepage (http://www.mysql.com/Manual_chapter/manual_Contrib.html#SEC455) und natürlich auf der CD zum Buch. Es ermöglicht Ihnen das komfortable Erstellen Ändern und Löschen von Tabellen, die Extraktion von CREATE TABLE-Direktiven aus bestehenden Tabellen, die Kontrolle der Server-Logfiles, das Abschicken von SQL-Queries in einem eigenen Fenster und vieles mehr. MyAccess ist ein Add-in. Um es zu installieren brauchen Sie lediglich die Datei MyAccess97.mda bzw. MyAccess2000.mda mit dem Add-in-Manager auszuwählen (EXTRAS | ADD-INS | ADD-IN-MANAGER | HINZUFÜGEN ...). Danach können Sie es durch Klick auf EXTRAS | ADD-INS | MYACCESS aufrufen und verwenden.

Der Zugriff auf die MySQL-Datenbank funktioniert jetzt nicht nur mit Microsoft Access sondern auch mit anderen Programmen, die auf ODBC-Datenquellen zugreifen können. Hierzu zählt z. B. die Tabellenkalkulation Microsoft Excel. Um mit Microsoft Excel auf eine ODBC-Datenquelle zugreifen zu können, müssen Sie Microsoft Query installieren. Microsoft Query ist ein optionaler Bestandteil von Microsoft Excel, in dem auch der Query-Assistent enthalten ist. Sollten sich Probleme bei der Installation von Microsoft Query ergeben, wenden Sie sich an Ihren Windows-Systemadministrator. Danach können Sie durch Klick auf DATEN | EXTERNE DATEN | NEUE ABFRAGE ERSTELLEN mit Hilfe des Microsoft Query Assistenten eine neue Abfrage erstellen und diese dann entweder im Microsoft Query weiter bearbeiten oder an Microsoft Excel "zurückgeben". Durch Rechtsklick irgendwo im Datenbereich des Spreadsheets können Sie die Datenbereichs-Eigenschaften verändern und z. B. dafür sorgen, daß die

Daten im Rechenblatt regelmäßig aktualisiert werden (z. B. bei jedem Öffnen der Datei).

Interessante Möglichkeiten bieten sich auch denjenigen, die elaboriertere statistische Auswertungen Ihrer Datenbestände durchführen wollen oder müssen und über das Statistikprogramm SPSS für Windows verfügen. Denn auch SPSS für Windows greift problemlos via ODBC auf die MySQL Datenbank zu. Und auch SPSS für Windows verwendet hierzu einen Assistenten (Sie bekommen ihn zu Gesicht, wenn Sie auf DATEI | DATENBANK ÖFFNEN | NEUE ABFRAGE ... klicken. Lassen Sie sich vom Assistenten leiten. Er produziert abschließend ein SPSS-Syntax-Skript, daß es Ihnen ermöglicht, zukünftig auf seine Dienste zu verzichten:

```
GET DATA
  /TYPE=ODBC
  /CONNECT=
    'DSN=Software-MySQL;'
    'DB=software;'
    'SERVER=db.ihr-server.de;'
    'UID=everyone;'
    'PWD=egal;'
    'PORT=3306;'
    'OPTION=0;'
    'STMT=;'
  /SQL ='SELECT T3.name AS name, T3.url AS url,'
        'T3.homepage AS homepage FROM soft T3'.
VARIABLE LABELS
  name "Programm-Name"
  url "Lokaler Download Url"
  homepage "Programm-Homepage".
```

Nachdem dieser Code ausgeführt wurde, ist all das möglich, was mit jeder anderen SPSS-Systemdatei auch möglich wäre (einschließlich des Abspeicherns als SPSS-Systemdatei).

9.3 Benutzerverwaltung

Das Zugriffskontrollsystem von MySQL bietet Ihnen sehr flexible und weitreichende Konfigurationsmöglichkeiten der Benutzerverwaltung. Dies schließt die Möglichkeit ein, sich problemlos und mit geringem Aufwand hoffnungslos ins digitale Nirwana zu konfigurieren. Die Benutzerverwaltung von MySQL ist nicht sehr intuitiv. In den MySQL-Mailinglisten kann man gelegentlich auch deutlichere Formulierungen lesen. Aus diesem Grund soll die MySQL-Benutzerverwaltung hier ausführlich ventiliert werden. Wir gehen zunächst davon aus, daß Sie noch keine Konfigurationsversuche nach dem try-and-error-Prinzip vorgenommen haben und sich die Privilegien-Tabellen noch in jungfräulichem Zustand befinden. Auf dieser Basis stellen wir Ihnen ein einfaches, aber im Regelfall ausreichend flexibles Konzept vor, mit dessen Hilfe sich problemlos Benutzer mit unterschiedlichen Rechten administrieren lassen. Wir verzichten dabei darauf, Ihnen verschiedene Möglichkeiten vorzuführen, die zum selben Ergebnis führen. So bietet z. B. `mysqladmin` ebenso einige Möglichkeiten, Einfluß auf die Benutzerverwaltung zu nehmen, wie die Standard SQL-Befehle `GRANT` und `REVOKE` mit ihren MySQL-Erweiterungen. Da `GRANT` aber erst mit der Version 3.22.11 implementiert und dem schon lange bestehenden System nur aufgesetzt wurde, erlaubt es einerseits keinen vollen Zugriff auf die Benutzerverwaltung (es können z. B. keine Benutzer vollständig entfernt werden), und es ist andererseits bei einfachen Operationen zwar leicht, bei komplexeren aber nur schwer verständlich. Wir zeigen daher hier die Benutzerverwaltung durch direkte Manipulation der Tabellen des Privilegien-Systems. Zum Ende des Abschnittes beschreiben wir einen Weg, wie Sie sich von einem vollständig zerstörtem Authentikationssystem befreien können, das Sie gar nicht mehr ins System läßt.

Im Zusammenhang mit der Erstinstallation und Basiskonfiguration von MySQL haben Sie auch das Skript `mysql_install_db` ausgeführt. Dieses Skript initialisiert die MySQL-Benutzerverwaltung, indem es eine Datenbank `mysql` anlegt, deren Aufgabe einzig und allein die Nutzerauthentikation ist. Die Datenbank `mysql` besteht aus sechs Tabellen, von denen für unser vereinfachtes Konzept lediglich zwei benötigt werden. Dies sind die Tabellen `user` und `db` (im übrigen sind dies auch die einzigen Tabellen, für die `mysql_install_db` bei der Erstinstallation Einträge vornimmt). Diese beiden Tabellen haben die folgende Struktur:

```
# phpMyAdmin MySQL-Dump
# http://phpwizard.net/phpMyAdmin/
#
# Host: localhost Datenbank : mysql
#
# -----
# Tabellenstruktur für Tabelle 'user'
#
CREATE TABLE user (
  Host char(60) NOT NULL,
  User char(16) NOT NULL,
  Password char(16) NOT NULL,
  Select_priv enum('N','Y') DEFAULT 'N' NOT NULL,
  Insert_priv enum('N','Y') DEFAULT 'N' NOT NULL,
  Update_priv enum('N','Y') DEFAULT 'N' NOT NULL,
  Delete_priv enum('N','Y') DEFAULT 'N' NOT NULL,
  Create_priv enum('N','Y') DEFAULT 'N' NOT NULL,
  Drop_priv enum('N','Y') DEFAULT 'N' NOT NULL,
  Reload_priv enum('N','Y') DEFAULT 'N' NOT NULL,
  Shutdown_priv enum('N','Y') DEFAULT 'N' NOT NULL,
  Process_priv enum('N','Y') DEFAULT 'N' NOT NULL,
  File_priv enum('N','Y') DEFAULT 'N' NOT NULL,
  Grant_priv enum('N','Y') DEFAULT 'N' NOT NULL,
```

```

References_priv enum('N','Y') DEFAULT 'N' NOT NULL,
Index_priv enum('N','Y') DEFAULT 'N' NOT NULL,
Alter_priv enum('N','Y') DEFAULT 'N' NOT NULL,
PRIMARY KEY (Host, User)
);

# -----
#
# Tabellenstruktur für Tabelle 'db'
#
CREATE TABLE db (
    Host char(60) NOT NULL,
    Db char(32) NOT NULL,
    User char(16) NOT NULL,
    Select_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Insert_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Update_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Delete_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Create_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Drop_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Grant_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    References_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Index_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    Alter_priv enum('N','Y') DEFAULT 'N' NOT NULL,
    PRIMARY KEY (Host, Db, User),
    KEY User (User)
);

```

Wie sich unschwer erkennen läßt, ähneln die Tabellen einander: Die ersten drei Spalten definieren jeweils, auf was sich die nachfolgenden Rechte-Spalten beziehen. In der user-Tabelle werden Rechte für Benutzer `User` vergeben, die von einem Host `Host` kommen und sich mit einem Kennwort `Password` identifiziert haben. In der User-Tabelle werden globale Rechte vergeben. Rechte, die Sie einem Benutzer in dieser Tabelle erteilen, können Sie ihm nicht in einer der anderen Tabellen wieder ent-

ziehen! In der user-Tabelle sollten daher nur DBMS-Administratoren Rechte erteilt bekommen! Mit DBMS-Administratoren meinen wir Personen, die berechtigt sein sollen, das DBMS insgesamt zu administrieren, also z. B. auch berechtigt sind, MySQL herunterzufahren, neue Datenbanken anzulegen, Optimierungen durch Veränderungen von Systemparametern vorzunehmen usw. Demgegenüber sind Datenbankadministratoren für die Administration einzelner Datenbanken zuständig und erhalten keine Privilegien, die es ihnen ermöglichen würde, das DBMS insgesamt zu gefährden.

In der db-Tabelle werden Benutzern `User` (deren Authentizität durch Vergleich des beim Anmelden angegebenen mit dem in der user-Tabelle hinterlegten Paßworts gesichert werden konnte), die von einem Host `Host` kommen, Rechte für eine Datenbank `Db` erteilt. Der Katalog der Rechte ist in der Db-Tabelle etwas kleiner, weil bestimmte Rechte, wie z. B. das Herunterfahren des DBMS, technisch nicht auf eine einzelne Datenbank beschränkt werden können.

Die Privilegien-Spalten (`Select_priv` usw.) der Tabellen können nur zwei Werte aufnehmen (Beim Spaltentyp `enum` werden die zulässigen Werte mit der CREATE TABLE-Anweisung abschließend definiert): `N` für No, Recht verweigert und `Y` für Yes, Recht erteilt. Voreingestellt ist der Wert `N`; Sie brauchen daher mit einem INSERT-Befehl nur für *die* Rechte-Spalten Werte setzen ('Y'), für die Sie Rechte erteilen wollen. Alle nicht ausdrücklich genannten Rechte werden vorenthalten (weil die Spalten mit 'N' vorbelegt sind). Die Rechte, die hier vergeben bzw. vorenthalten werden können, beziehen sich auf die entsprechenden SQL-Befehle. Ein 'Y' in der Spalte `Select_priv` erlaubt dem Benutzer, beliebige SELECT-Statements abzuschicken, der gleiche Eintrag in der Spalte `Insert_priv` erlaubt INSERT-Statements usw.

Das Skript `mysql_install_db`, daß die Benutzerverwaltung initialisiert hat, hat (nach dem es mit den für die Tabellen `User` und `Db` schon gezeigten `CREATE TABLE` Befehlen zunächst die Tabellen angelegt hat) folgende `INSERT`-Befehle an den Server geschickt:

```
# Legt den DBMS-Administrator root mit "OhnePaßwort" an
INSERT INTO user VALUES ('localhost','root','','Y','Y',
    'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y');
INSERT INTO user VALUES ('$hostname','root','','Y','Y',
    'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y');

# Legt einen User "OhneNamen" mit "OhnePaßwort" an
INSERT INTO user VALUES ('localhost','','','N','N','N',
    'N','N','N','N','N','N','N','N','N','N','N');
INSERT INTO user VALUES ('$hostname','','','N','N','N',
    'N','N','N','N','N','N','N','N','N','N','N');

# Gibt "OhneNamen" alle Rechte an der Datenbank test
# sowie an allen Datenbanken, die mit test_ beginnen.
INSERT INTO db VALUES ('%','test','','Y','Y','Y','Y',
    'Y','Y','N','Y','Y','Y');
INSERT INTO db VALUES ('%','test\_%',','','Y','Y','Y','Y',
    'Y','Y','N','Y','Y','Y');
```

Die Variable `$hostname` wird dabei vom Shell-Skript `mysql_install_db` durch den Hostnamen Ihres Systems ersetzt. Dies führt an sich zu einem nicht sonderlich sinnvollen Doppeleintrag (`localhost` ist schließlich nur eine Art Alias für Ihren Hostnamen). Dies ist eine Absicherung für den Fall, daß ihr System MIT-pthreads verwendet. In diesem Fall ist keine Verbindungsaufnahme via `localhost` möglich (`localhost`-Verbindungen werden über UNIX Sockets realisiert, die von MIT-pthreads nicht unterstützt werden). Wenn Sie den `mysql`-Client ohne Angabe der Option `--host=ihre.domain.de` aufrufen konnten, können Sie die Zeilen mit `$hostname` in der `User`-Tabelle getrost löschen. Und wenn Sie schon einmal beim Löschen sind, löschen Sie auch

gleich die Einträge für den Benutzer "OhneNamen" (und zwar sowohl in der User- als auch in der Db-Tabelle). Erstens dürfte es nicht in Ihrem Interesse sein, das unbekannte Benutzer nach Belieben Ihr DBMS benutzen. Zweitens dürften die Mehrzahl der access-denied-Probleme beim Zugriff auf MySQL mit der Benutzung von Wildcards zusammenhängen (weil MySQL durch eine eigenwillige, menschlichen Denkgewohnheiten widersprechenden Bevorzugung von allgemeineren vor spezifischeren Benutzerangaben für Verwirrung sorgt). Vermeiden Sie Wildcards, soweit möglich und solange Ihnen nicht vollständig klar ist, wie MySQL diese behandelt. Wenn Sie Wildcards in der Host-Spalte verwenden, schreiben Sie immer auch einen zweiten identischen Tabelleneintrag für localhost (bzw. für ihre.domain.de auf MIT-pthread-Systemen).

```
shell> mysql -u root
mysql> DELETE FROM user WHERE Host="ihre.domain.de";
mysql> DELETE FROM user WHERE User='';
mysql> DELETE FROM db;
```

Nun sieht Ihre Benutzerverwaltung schon wesentlich übersichtlicher aus. Lediglich ein Eintrag (der für den DBMS-Administrator root von localhost aus) ist übrig geblieben. Und Übersichtlichkeit und Transparenz in der Benutzerverwaltung schützt Sie vor Datenverlusten. Der Eintrag für root ist noch nicht durch ein Kennwort abgesichert, was wir nun schleunigst nachholen:

```
mysql> UPDATE user SET Password = PASSWORD('top_secret')
    -> WHERE User='root';
mysql> FLUSH PRIVILEGES;
```

Damit ist erstens die Benutzerverwaltung von den ungünstigen Voreinstellungen gesäubert, zweitens der root-Account mit einem Kennwort geschützt und drittens klar, wie DBMS-Administratoren angelegt werden müssen. Bei Bedarf können Sie weitere Benutzer zu DBMS-Administratoren durch einen entsprechenden INSERT-Befehl auf die User-Tabelle anlegen. Die Direk-

tive `FLUSH PRIVILEGES`; sorgt dafür, daß MySQL die `mysql`-Tabellen neu einliest. Ohne diese Direktive bleiben alle Ihre Bemühungen ergebnislos (ein vergessenes `FLUSH PRIVILEGES` hat schon so manchen angehenden Datenbankadministrator stundenlang beschäftigt gehalten. Die Funktion `PASSWORD()` sorgt für die Verschlüsselung des Kennworts. Sollten Sie diese Funktion vergessen und `SET Password = 'top_secret'` angeben, wird das Kennwort unverschlüsselt abgespeichert. Dies führt dann dazu, daß später eine Authentifizierung scheitert (weil das bei der Authentifizierung erfragte Kennwort zunächst verschlüsselt wird und erst der verschlüsselte Ausdruck mit dem unverschlüsselten Eintrag in der `user`-Tabelle verglichen wird).

In aller Regel ist es das Sinnvollste (und für Sie als DBMS-Administrator das Transparenteste), wenn alle weiteren Benutzer (einschließlich möglicher Datenbankadministratoren) Rechte datenbankbezogen zugeteilt bekommen. Dabei ist folgendermaßen vorzugehen: Zunächst werden weitere Benutzer in der `user`-Tabelle angelegt. Bei diesen setzen Sie lediglich Werte für die Spalten `Host`, `User`, `Password`. Sie erstellen damit Benutzer, die keinerlei globalen Rechte haben (denn die Privilegien-Spalten sind ja mit 'N' vorbelegt). Diesen Nichtskönnern erteilen Sie nun gezielt einzelne Rechte für einzelne Datenbanken, indem Sie entsprechende Einträge in der Tabelle `db` vornehmen. Vergeben Sie stets nur so viele Rechte, wie unbedingt notwendig! Jemand, der fehlerhafte Eintragungen in der Adresdatenbank korrigieren oder Adressen ergänzen soll, braucht nicht das Recht, Tabellen zu erstellen oder zu löschen. Fragen Sie sich auch, ob Sie dem Datenbankadministrator der Adresdatenbank (der auch die Tabellenstruktur ändern können soll) wirklich `GRANT`-Rechte erteilen wollen (und ihm damit die Möglichkeit geben, seine Rechte beliebig an andere zu vererben). Nehmen wir also an, Sie hätten eine Datenbank `adressen`. In dieser Datenbank soll der Benutzer `mueLLer` Einträge ändern, löschen und hinzufügen dürfen. Der Benutzer `supermueLLer` soll Daten-

bankadministrator werden und zusätzlich das Recht erhalten, Tabellen anzulegen, zu ändern und zu löschen. Die folgenden SQL-Statements erledigen das:

```
# ein rechtloser User mueller wird eingerichtet
mysql> INSERT INTO user
    -> SET Host = 'localhost',
    ->     User = 'mueller',
    ->     Password = PASSWORD('muellerpwd');
# ... und erhält eingeschränkte Rechte für die
# Datenbank adressen
mysql> INSERT INTO db
    -> SET Host = 'localhost',
    ->     Db = 'adressen',
    ->     User = 'mueller',
    ->     Select_priv = 'Y', Insert_priv = 'Y',
    ->     Update_priv = 'Y', Delete_priv = 'Y';
```

In gleicher Weise richten Sie nun noch den Datenbankadministrator `supermueller` ein und ergänzen seine Rechte (`Create_priv = 'Y'`, `Drop_priv = 'Y'`, `References_priv = 'Y'`, `Index_priv = 'Y'`, `Alter_priv = 'Y'`). Den Eintrag `References_priv` haben Sie nur schon einmal vorsichtshalber vorgenommen. Derzeit bewirkt er noch nichts. Wenn Sie diese Adressdatenbank über das Intranet allen Mitarbeitern zum lesenden Zugriff zur Verfügung stellen wollen (z. B. via PHP, vgl. Kapitel 13.2), dann legen Sie zusätzlich noch einen Benutzer `everyone` an (ohne Rechte in der user-Tabelle) und mit Zugriffsrecht aus dem IP-Bereich Ihres Firmennetzwerkes (z. B. `'190.120.200.%'`). Vergeben Sie entweder ein allgemeines Paßwort ('anonymous') oder lassen Sie die Password-Spalte ohne Eintrag (Dann darf beim Einloggen auch keins angegeben werden!). Wenn Sie im Host-Feld Wildcards verwenden, benutzen

Sie keine Domain-Namen, sondern (aus Sicherheitsgründen) IP-Nummern.

Wenn Sie den Wunsch verspüren oder sich bei Ihnen die Notwendigkeit ergibt, das vorgestellte System der Benutzerauthentifizierung zu verfeinern, so bietet MySQL Ihnen weitreichende Möglichkeiten, die hier nicht im Einzelnen vorgestellt werden können. Wenn Es Ihnen sinnvoll erscheint, können Sie unterschiedliche Rechte für die gleichen Benutzer definieren, je nach dem, ob ein Zugriff aus dem Firmennetz oder von Sonstwoher erfolgt. Sie können Rechte nicht nur für Datenbanken sondern auch für einzelne Tabellen und auch für einzelne Spalten in Tabellen gewähren oder verweigern. Yves Carlier ist es gelungen, die Angelegenheit so kompliziert zu gestalten, daß es ihm angebracht erschien, eigens ein Tool zu programmieren, daß ihn bei der Diagnose der von ihm erstellten Zugriffsberechtigungen unterstützt. Das Tool ist Bestandteil der MySQL-Distribution. Informationen zu seiner Verwendung erhalten Sie durch Aufruf von `mysqlaccess --help`.

Zum Schluß geben wir noch die versprochene Hilfestellung für den Fall, daß Sie sich um Kopf und Kragen konfiguriert haben und jetzt überhaupt nicht mehr auf den Datenbankserver zugreifen können. Das Prinzip ist einfach: Datenbank-Server herunterfahren, Datenbank-Server mit der Option `--skip-grant-tables` starten (das deaktiviert das gesamte Zugriffskontrollsystem), `mysql`-Tabellen aufräumen und DBMS-Administrator einrichten, Datenbank-Server wieder herunterfahren und regulär (mit Zugriffskontrollsystem) neustarten, fertig. Die Pfade müssen Sie natürlich ggf. anpassen.

```
# Zunächst wird der DB-Server heruntergefahren
shell> kill `cat /var/lib/mysql/hostname.pid`
# Neustart des DB-Servers ohne Rechtetabellen
shell> /usr/local/mysql/libexec/mysqld --skip-grant-tables
```

```

shell> mysql
mysql> use mysql;
# Löschen der bestehenden einträge für root
mysql> DELETE FROM user WHERE User='root';
mysql> DELETE FROM db WHERE User='root';
mysql> DELETE FROM tables_priv WHERE User='root';
mysql> DELETE FROM columns_priv WHERE User='root';
# Anlegen eines DBMS-Administrators root mit dem
# Kennwort "top_secret"
INSERT INTO user VALUES ('localhost','root',
    PASSWORD('top_secret'),'Y','Y','Y','Y',
    'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y');
mysql> exit
# DB-Server abschießen und neustarten
shell> kill `cat /var/lib/mysql/hostname.pid`
shell> /usr/local/bin/safe_mysqld &

```

9.4 Datensicherung und Wartung

Sicherlich sind Ihnen die drei wichtigsten Regeln im Umgang mit EDV geläufig? Sie seien hier noch einmal in Erinnerung gerufen:

1. Erstelle in regelmäßigen Abständen Sicherheitskopien!
2. Erstelle in regelmäßigen Abständen Sicherheitskopien!
3. Erstelle in regelmäßigen Abständen Sicherheitskopien!

Zur Datensicherung haben Sie bei MySQL mehrere Möglichkeiten. Das einfachste ist es eine Sicherung auf Dateiebene vorzunehmen: MySQL speichert sämtliche relevanten Informationen einer Datenbank in einem Verzeichnis `DATADIR/dbname` (z. B.: `/var/lib/mysql/adressen`). Als Datensicherung ist es völlig ausreichend, diese Verzeichnisse an einen sicheren Ort zu kopieren. Um ein konsistentes Backup zu gewährleisten, müssen Sie

allerdings für die Dauer der Datensicherung den Datenbank-Daemon herunterfahren. Sie können auch das gesamte Datenverzeichnis z. B. mit folgendem Befehl komprimiert abspeichern:

```
> tar zcvf backup-2000.07.tar.gz /var/lib/mysql
```

Wenn Sie das Backup eines Tages benötigen, kopieren Sie einfach die gesicherten Datenbank-Verzeichnisse zurück in das Verzeichnis DATADIR. Auf diesem Wege können übrigens auch problemlos Duplikate ganzer Datenbanken (z. B. zu Testzwecken) angelegt werden (`cp -R /var/lib/mysql/dbname /var/lib/mysql/dbname-duplikat`; denken Sie an die korrekte Vergabe der Zugriffsrechte auf Dateiebene, ggf. ist ein `chown` erforderlich).

Eine andere Möglichkeit ist die Erstellung eines SQL-Dumps. Wenn Sie beim Aufruf von `mysqldump` die Option `--lock-tables` angeben (bzw. `--opt`, die Option `--opt` beinhaltet `--lock-tables`), brauchen Sie sich um die Konsistenz Ihres Backups keine Sorgen zu machen; `mysqldump` sorgt vor dem Lesen einer Tabelle für ein `LOCK TABLES SQL`-Statement. Dies verhindert während des Lesevorganges gleichzeitige `UPDATE`-Versuche durch dritte. Beispiel:

```
> mysqldump -u root -p --opt --all-databases \  
> |gzip>backup2000.07.sql.gz
```

Der Aufruf führt ein Vollbackup aller Datenbanken durch und speichert dieses komprimiert in der Datei `backup2000.07.sql.gz`. Dabei legt MySQL durchaus eine erträgliche Geschwindigkeit an den Tag: Die oben genannte Befehlsfolge benötigt (einschließlich der Komprimierung) für 52MByte Daten auf dem Referenzsystem knapp 15 Sekunden (und die erzeugte Datei verbraucht 2,9Mbyte Speicherplatz). Ohne Komprimierung verkürzt sich die Zeit auf 10 Sekunden. Zurückgespielt wird eine so erstellte Sicherheitskopie durch den Aufruf:

```
> mysql -u root -p <backup2000.07.sql
```

Eine sehr elegante Möglichkeit, auf diese Weise erstellte Vollbackups durch inkrementelle Differenzbackups zu ergänzen, stellt das Update-Log Feature von MySQL dar. Wenn Sie den Datenbankdaemon mit der Option `--log-update=dateiname` starten (bzw. in der Datei `/etc/my.cnf` `log-update=dateiname` eintragen, und anschließend ein `mysqladmin -u root -p flush-logs` durchführen), wird im Daten-Directory von MySQL eine Datei angelegt, in der jeder SQL-Befehl abgespeichert wird, der die Datenbank in irgendeiner Weise verändert. Nicht gesichert werden z. B. alle SELECT Befehle. Erfolgreiche UPDATE oder DELETE Statements (die z. B. in der WHERE-Clausel eine Bedingung stellen, die auf keine Zeile zutrifft) werden ebenfalls nicht abgespeichert. Ihr Update-Log-Datei enthält also ausführbare SQL-Statements, die Sie - wie SQL-Datenbankdumps - problemlos wieder zurückspielen können. Bei jedem Neustart des Datenbankservers (ebenso wie nach jedem `mysqladmin refresh` oder `mysqladmin flush-logs` Aufruf oder nach jedem FLUSH LOGS Statement) wird ein neues Update-Log-File angelegt (die erste Update-Log-Datei erhält den Dateinamen `dateiname.1`, die zweite den Namen `dateiname.2` usw.). Wenn Sie ein Backup aus Update-Log-Dateien durchführen wollen, geben Sie z. B. an:

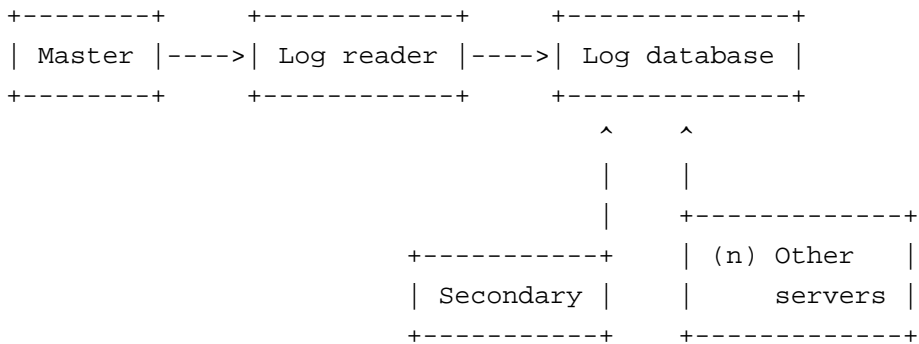
```
> ls -l -t -r dateiname.[0-9]* | xargs cat | mysql
```

Der Aufruf übergibt die Update-Log Dateien in der richtigen Reihenfolge (beginnend mit der ältesten) an den Datenbankserver.

Nach einem Daten-Gau, der sich auch mit `myisamchk` (s. u.) nicht beheben läßt (was allerdings relativ unwahrscheinlich ist), brauchen Sie also nichts anderes zu tun, als Ihr letztes Vollbackup einzuspielen und anschließend für den Zeitraum seit Durchführung dieses Vollbackups die entsprechenden Update-Log-Dateien an den Server zu schicken. Die Update-Log-Dateien können Sie übrigens auch verwenden, wenn Sie Ihre Datenbank aus Sicherheits- oder Performance Gründen auf einen

zweiten Host dupliziert haben. In einem solchen Szenario verwenden Sie die Update-Log-Files der Master-Datenbank als Replikations-Skripts für den Mirror (und rufen diese z. B. durch einen Cron-Job in den in Ihrem Fall sinnvoll erscheinenden Zeitintervallen auf). Wenn Sie ernsthaft über eine Datenbankreplikation nachdenken, werfen Sie mal einen Blick auf das MySQL Clustering Toolkit des Prometheus Project (http://prometheus.zerodivide.net/mysql_cluster/), dem wir das folgende ASCII-Kunstwerk entnommen haben:

How Prometheus MySQL Clustering Toolkit works(ASCII art):



Auf Ihre Backups werden Sie mit einiger Wahrscheinlichkeit nur zurückgreifen müssen, wenn Sie entweder selbst versehentlich oder aber andere Ihnen absichtlich Ihre Daten gelöscht haben. Bei Unglücksfällen wie "abgeschmierten" Servern oder Stromausfällen reicht fast immer die Verwendung des MySQL-Wartungs- und Rettungstools `myisamchk` (für Tabellen im neuen Standardformat MyISAM; für Tabellen im alten ISAM-Format verwenden Sie `isamchk`). Falls Sie sich nicht sicher sind, welches Format Ihre Tabellen haben: im MySQL Daten-Verzeichnis (z. B.: `/var/lib/mysql`) finden sich in den Unterverzeichnissen `dbnamen` für Tabellen im MyISAM-Format Dateien mit den Endungen `.MYI`

und .MYD und für Tabellen im ISAM-Format Dateien mit den Endungen .ISM und .ISD). Sie können gleichzeitig Tabellen in unterschiedlichen Format innerhalb einer Datenbank vorhalten. Es ist aber sinnvoll (und ohne großen Aufwand auch möglich) die Tabellen einheitlich in das neue Format MyISAM zu bringen, weil MyISAM gegenüber ISAM eine ganze Reihe von Vorzügen hat: Unter anderem liegen die Tabellen jetzt plattformübergreifend im gleichen Dateiformat vor. Man kann z. B. problemlos den Inhalt des MySQL-Daten-Verzeichnisses einer Linux-Maschine in das Datenverzeichnis eines Microsoft Windows Heimcomputers kopieren und darauf mit der Microsoft Windows Version von MySQL zugreifen. Verschiedene andere Veränderungen sorgen für deutliche Performancegewinne sowie für kleinere Tabellen. Mit dem Perl-Skript `mysql_convert_table_format` können Sie in einem Rutsch alle Tabellen einer Datenbank in das neue Format konvertieren:

```
> mysql_convert_table_format dbname --user=root \  
--password=top_secret
```

Es ist sinnvoll, in regelmäßigen Abständen alle Tabellen auf Fehler zu überprüfen. Sie können dies mit einem Cron-Job automatisieren:

```
0 3 * * * /usr/local/mysql/bin/myisamchk -s \  
/var/lib/mysql/*/*.MYI | \  
mail -s MySQL-Status db-admin@ihre-domain.de
```

Eine solche Zeile in Ihrer Crontab-Datei sorgt dafür, das allnächtens um 3:00 Uhr alle Tabellen überprüft werden. Das Ergebnis wird Ihnen als E-Mail zugestellt. Die Option `myisamchk -s` (silent) verhindert eine Ausgabe, wenn alles in Ordnung war. Wenn Sie den Datenbankserver üblicherweise mit der Option `--skip-locking` starten (müssen), dann müssen Sie vor der Überprüfung von Tabellen mit `myisamchk mysqld` herunterfahren (mindestens sollten Sie `mysqladmin flush-tables` durchführen). Wenn Sie mit `myisamchk` Reparaturen durchführen wollen,

dann ist es zwingend erforderlich, daß während des Reparaturvorgangs nicht auf die Tabellen zugegriffen wird. Sie können die Prüfläufe aber auch an einer Sicherheitskopie durchführen, es ist nicht erforderlich, daß sich die zu prüfenden Tabellen im MySQL-Datenverzeichnis befinden.

Wenn Sie `myisamchk` ohne Optionen aufrufen, führt das Programm eine Standard-Überprüfung der Tabelle durch, mit der (laut Handbuch) 99,99% aller Fehler gefunden werden. Anschließend gibt das Tool eine Meldung aus, der entnommen werden kann, ob Fehler aufgetreten sind und welche Prüfungen durchgeführt wurden:

```
# Es soll die Tabelle soft der Datenbank software
# geprüft werden
> myisamchk /var/lib/mysql/software/soft
Checking MyISAM file: soft
  Data records:      55   Deleted blocks:      0
  - check file-size
  - check key delete-chain
  - check record delete-chain
  - check index reference
  - check data record references index: 1
  - check record links
```

Wenn Sie Grund zu der Annahme haben, das ein Tabellenfehler vorliegt, `myisamchk` diesen aber nicht findet, verwenden Sie die option `-e` (extended). Bei großen Tabellen kann dies allerdings sehr lange dauern.

Laut Handbuch machen sich Tabellenfehler durch folgende Symptome bzw. Fehlermeldungen bemerkbar:

- `'tabellen_name.frm'` ist für Veränderungen gesperrt

- Kann Datei ' tabellen_name.MYI' nicht finden (Fehler: ###)
- Fehler ### (table handler)
- Unexpected end of file
- Record file is crashed

(Die letzten beiden Fehlermeldungen finden sich nicht in der englischen Version der Datei `errmsg.txt` und sind daher vermutlich fest im Programm eincompiliert). Wenn solche Symptome auftreten, müssen die Tabellen repariert werden. Das Handbuch empfiehlt hierbei folgende Vorgehensweise:

Schritt 1: Tabellenprüfung. Prüfen Sie die Tabellen mit `myisamchk *.MYI` oder `myisamchk -e *.MYI` (wenn Sie viel Zeit haben). Reparieren Sie nur die Tabellen (entsprechend der Angaben unter Schritt 2), für die `myisamchk` Fehlermeldungen ausgegeben hat. Wenn Sie bei der Prüfung mysteriöse Fehlermeldungen erhalten (z. B. out-of-memory-Meldungen) oder wenn sich `myisamchk` beim Prüfen aufhängt, versuchen Sie es mit Schritt 3.

Schritt 2: Einfache, sicherere Reparatur. Versuchen Sie es zunächst mit `myisamchk -r -q problemtabelle` (`-r -q` steht für "quick recovery", schnelle Wiederherstellung). `myisamchk` versucht dann, die Index-Datei zu reparieren, ohne die Daten-Datei zu verändern. Dies funktioniert, solange die Datentabelle alle notwendigen Informationen enthält. Bleibt dieser Versuch erfolglos, gehen Sie wie folgt vor:

1. Erstellen Sie ein Backup der Daten-Datei (das ist die mit der Endung `.MYD`) bevor Sie fortfahren.
2. Verwenden Sie `myisamchk -r problemtabelle` (`-r` steht für "recovery", Wiederherstellung). Dies löscht fehlerhafte

und zerstörte Datensätze und rekonstruiert die Index-Datei.

3. Wenn auch dieser Versuch fehlschlägt, versuchen Sie es mit `myisamchk -safe-recover problemtabelle`. Diese Option verwendet ein Rekonstruktionsverfahren, das langsamer ist, aber dafür einige Probleme beheben kann, die mit dem Standardverfahren nicht gelöst werden können.

Wenn all das nichts gefruchtet hat und Sie weiterhin mysteriöse Fehlermeldungen erhalten (out of memory), versuchen Sie Schritt 3.

Schritt 3: Schwierige Reparatur. In Ihrem Fall ist es notwendig, ein neue Index-Datei zu erstellen. Verfahren Sie wie folgt:

1. Verschieben Sie die Datendatei an einen sicheren Ort.
2. Verwenden Sie die Tabellenbeschreibungs-Datei um eine neue, leere Datendatei und eine neue Index-Datei zu erzeugen, indem Sie den MySQL-Client aufrufen und den Inhalt der betreffenden Tabelle löschen (`DELETE FROM problemtabelle`).
3. Kopieren Sie die alte Datendatei über die gerade neu erstellte (Verschieben Sie die alte Datei nicht bloß, sondern erzeugen Sie eine Kopie. Damit Sie eine Sicherheitskopie Ihrer Daten haben, falls etwas schief geht).

Gehen Sie zurück zu Schritt 2. `myisamchk -r -q` sollte nun das Problem beheben können (Es ist freilich nicht unsere Absicht, Sie durch diese Empfehlung in einer Endlos-Schleife auf Dauer beschäftigt zu halten).

Schritt 4: Sehr schwierige Reparatur. Bis hierhin sollten Sie nur gelangen, wenn auch die Datenbeschreibungs-Datei defekt ist (das ist die mit der Endung `.frm`). Eigentlich sollte ein solcher Fall niemals auftreten, weil die Datenbeschreibungs-Datei

nach der Tabellenerstellung nicht mehr verändert wird. Haben Sie sich an wenigstens eine der drei wichtigsten Regeln gehalten, die wir Ihnen zu Beginn des Abschnitts beim Umgang mit EDV anempfohlen haben? Dann bereitet Ihnen auch dieser "worst case" keine größeren Probleme.

1. Restaurieren Sie die Datenbeschreibungs-Datei von Ihrem Backup-Archiv und gehen Sie zurück zu Stufe 3. Sie können gleichzeitig auch die Index-Datei restaurieren. Dann setzen Sie Ihre Bemühungen bei Stufe 2 fort (verwenden Sie in diesem Fall `myisamchk -r`).
2. Sollten Sie über keine Sicherheitskopie verfügen, dafür aber über ein so gutes Gedächtnis, daß es Ihnen möglich ist, die Tabellenstruktur exakt zu rekonstruieren, bleibt Ihnen eine letzte Möglichkeit: Erstellen Sie eine leere Kopie der Tabelle in einer anderen Datenbank (z. B. in der Datenbank `test`). Löschen Sie die neue Daten-Datei und verschieben Sie die Datenbeschreibungs- und die Index-Datei von der anderen Datenbank in ihre defekte Datenbank. Das Verfahren erstellt eine neue Datenbeschreibungs- und eine neue Index-Datei, läßt aber Ihre Daten-datei ungeschoren. Gehen Sie nun zurück zu Schritt 2 und versuchen Sie, die Index-Datei zu rekonstruieren.

Wenn auch das nicht hilft, ist endlich der Zeitpunkt gekommen, an dem sich der Aufwand bezahlt gemacht hat, den Sie mit der Eingangs dargelegten Sicherungsstrategie auf sich genommen haben. Mit Hilfe regelmäßiger Komplettsicherungen und dem Update-Log-Feature von MySQL können Sie das Risiko von Datenverlusten extrem gering halten.