

INTERNET

PROFESSIONELL

**Workshop
Navigation**
Orientierung
mit Bread-
crumbs
>S.68

10 Jahre Jubiläum-Ausgabe

Datenbank-Tricks:

MySQL

>S.26, 82

- > Komfortabel einrichten und verwalten
- > Performance und Sicherheit optimieren
- > PHP 5.1: geniale neue Datenbank-Features

VOLLVERSIONEN
auf Heft-CD



Site-Builder

Professionelle Software zum Erstellen & Gestalten von Websites am Server alle Details: >S. 20

Opera 8.5



Voice over IP

Die besten Tools und Programme zum VoIP

PHP: Gratis-Code

>S.88

- > Fremde Features in eigene Skripts einbauen
- > Klassen nutzen statt selbst programmieren

Megatrend Ajax

>S.92

- > Daten dynamisch per Javascript nachladen
- > mit Einsteiger-Workshop

PLUS

- Workshop: Privacy-Policy für Ihre Website S.56
- Javascript-Praxis: Lexikon-Zugriff per Link S.72
- Spam-Bekämpfung: externe Antispam-Dienstleister S.48





MySQL-Datenbanken einrichten, programmieren und optimieren

Dynamik für Websites

Wer eine Datenbank für seine Website braucht, dessen Wahl fällt in den meisten Fällen auf MySQL. Die Datenbank ist kostenlos, auf die Bedürfnisse im Internet zugeschnitten, fast jeder Provider bietet MySQL-Datenbanken an, und außerdem ist MySQL besonders einfach über PHP anzusprechen. **Von Max Bold**

» Wer eine dynamische Website aufsetzen will, muss sich zuerst nach einer passenden Umgebung dafür umsehen. In vielen Fällen wird dies eine klassische Lamp-Umgebung sein. Die gibt es im Angebot der Provider in zahlreichen Spielarten.

Für einfache dynamische Web-Applikationen wird in den meisten Fällen ein Shared-Hosting-Paket genügen. Die meisten Provider bieten teilweise schon für wenig Geld eine solche Konstellation aus Linux, Apache, PHP und der Datenbank MySQL an.

Im unteren Preissegment wird sich ein Mieter eines solchen Pakets in der Regel mit einer vorkonfigurierten Datenbank begnügen müssen, in der er dann die für seine Anwendungen erforderlichen Tabellen einrichten kann. Dies stellt jedoch nur ein Grundgerüst mit zahlreichen Einschränkungen dar.

Für zahlreiche PHP-Skripts und -Applikationen ist die Möglichkeit, eine eigene Datenbank anlegen zu können, Voraussetzung. Jeder Versuch, ein solches Skript auf einem Provider-Paket zu installieren, das diese Option nicht bietet, muss daher scheitern. Auch die in höheren Preissegmenten angebotene größere Zahl an Datenbanken hilft hier nicht weiter, wenn deren Namen fest und unveränderbar vorgegeben sind.

Ein weiteres Kriterium, das bei der Auswahl eines solchen Shared-Hosting-Pakets beachtet werden muss, ist natürlich die Version des installierten MySQL-Servers. Zahlreiche Skripts stellen hier klare Anforderungen. Um auf der sicheren Seite zu sein, sollten Sie auf einer aktuellen Version bestehen.

Ein weiteres Detail, das auf Ihrer Checkliste erscheinen sollte, sind eventuelle Limits beim Speichervolumen (Quota) von Datenbanken. Entweder gibt es hier ein festes Limit pro Datenbank, oder der für Datenbankanhalte verbrauchte Speicherplatz wird mit dem gesamten Web-space, der zu dem Tarifpaket gehört, verrechnet.

Ein weiterer Punkt, den Sie beachten müssen, ist die konkrete Ausgestaltung der Implementierung des Datenbankservers. In vielen Fällen gelten bei Shared-Hosting-Paketen Sicherheitseinschränkungen. So sind vielfach die MySQL-Server nur direkt über die eigene Präsenz erreichbar und zum Beispiel nicht per ODBC von Ihrem lokalen Rechner aus. Dies dient zum Schutz vor Angriffen von außen; macht jedoch die Installation eines Administrations-Tools zur Datenbankwartung auf dem lokalen Rechner unmöglich. Manche Provider bieten hier die Möglichkeit, dass der Mieter den externen Datenbank-Zugriff auf eigenes Risiko freischalten kann.

»Datenbank-Alternativen«

Neben MySQL gilt die Datenbank PostgreSQL, ebenfalls aus dem Open-Source-Lager, als Geheimtipp. Allerdings ist diese Alternative bei den Shared-Hosting-Angeboten nicht gerade häufig vertreten.

Auch die Kombination von Windows-Hosting und MySQL gibt es. Allerdings eilt dieser Variante der Ruf voraus, weniger stabil und performant zu sein als ein reines Lamp-System. Die Devise sollte hier lauten: wenn schon Microsoft, dann komplett. Die Kombination aus Windows und den von Microsoft angebotenen Datenbanken Access und MS SQL ist hier in jedem Fall die bessere Wahl.

»Root-Server mit allen Freiheiten«

Wem ein solches Shared-Hosting-Paket nicht genügt, der kann sich einen dedizierten Server auf Lamp-Basis mieten. Dank Root-Zugriff bietet diese Variante ungleich mehr Rechte und vielfältige Gestaltungsmöglichkeiten. Sie können Ihren MySQL-Server individuell konfigurieren, beliebig viele Datenbanken anlegen und das Quota ist nur durch die verfügbare Festplattenkapazität begrenzt.

Hier haben Sie natürlich die Option, eine andere Open-Source-Datenbank wie zum Beispiel PostgreSQL oder ein

kommerzielles Produkt zu installieren. Bei MySQL haben Sie auf einem eigenen Server auch die Möglichkeit, jederzeit selbst Updates und Patches einzuspielen und den Datenbank-Server individuell zu konfigurieren. Da Sie hier auch vollen Zugriff auf die User-Verwaltung haben, stehen Ihnen ungleich mehr Optionen bei der Anlage von Usern und bei der Vergabe von Zugriffsrechten zur Verfügung. Für kundige Administratoren ist dies ein Dorado, es bringt allerdings auch einige Sicherheitsrisiken mit sich.

Als Fazit kann man festhalten, dass das Datenbank-Angebot beim Shared Hosting mit zahlreichen Limits verbunden ist, nicht nur bei der Performance, sondern vor allem auch bei der Administration. Komplexe Applikationen lassen sich damit kaum realisieren, und als Basis für interaktive Web-Auftritte mit hoher Besucherfrequenz sind sie ebenfalls wenig geeignet. Die einzige realistische Alternative sind hier dedizierte Server.

»MySQL programmieren«

PHP versteht sich dank der integrierten Schnittstellen mit MySQL sehr gut. Es stehen zwei Varianten für die Programmierung zur Verfügung: die klassische *mysql*-Methode und die modernere, objektorientierte *mysqli*-Methode, die in

Info

MySQL-Benutzerverwaltung

Der MySQL-Server verwaltet die Berechtigungen in verschiedenen Systemtabellen in der MySQL-Datenbank. Dieser Kasten zeigt, welche Informationen in den Tabellen stecken.

■ Tabelle *user* – hier werden Benutzer und Passwörter verwaltet und globale Berechtigungen für alle Datenbanken gepflegt.

■ Tabelle *db* – Benutzer, die uneingeschränkte Berechtigungen auf alle Tabellen einer Datenbank haben, werden in dieser Tabelle abgelegt.

■ Tabelle *host* – anhand des Hostnamens des Clients bestimmt diese Tabelle die Berechtigungen auf alle Tabellen einer Datenbank.

■ Tabelle *tables_priv* – hiermit werden Benutzerrechte gezielt einzelnen Tabellen einer Datenbank zugeordnet.

■ *columns_priv* – hiermit werden spezifische Rechte für einzelne Spalten einer Tabelle bestimmt.



Bei Shared Hosting erfolgt die Datenbank-Konfiguration über ein Web-Interface.

Das Administrations-Tool PhpMyAdmin bietet zahlreiche Exportfunktionen.

Info

Auf einen Blick

- » Der Artikel zeigt, was Sie beim Arbeiten mit PHP und MySQL beachten müssen.
- » Er zeigt die Unterschiede der verschiedenen MySQL-Implementierungen bei den einzelnen Hosting-Angeboten.
- » Außerdem erfahren Sie, wie Sie mit Admin-Tools wie PhpMyAdmin MySQL-Datenbanken komfortabel verwalten können.

Info

MySQL-Benutzerrechte

Die Berechtigungen eines MySQL-Users entscheiden über seine Möglichkeiten im Umgang mit dem Server.

Die folgende Übersicht zeigt, welche Möglichkeiten sich im Detail hinter den einzelnen Rechten verbergen. Für den kompletten MySQL-Server, einzelne Datenbanken oder Tabellen sind die folgenden Berechtigungen möglich:

- **ALTER** – erlaubt die ALTER-Anweisung
- **CREATE** – erlaubt das Erstellen neuer Tabellen
- **DELETE** – erlaubt das Löschen von Datensätzen
- **DROP** – erlaubt das Löschen von Tabellen
- **INDEX** – erlaubt das Arbeiten mit Indizes
- **INSERT** – erlaubt das Anlegen neuer Datensätze
- **SELECT** – erlaubt das Erstellen von Abfragen
- **UPDATE** – erlaubt das Verändern von Datensätzen

Für die einzelnen Spalten einer Tabelle sind die Berechtigungen **ALL**, **SELECT**, **INSERT** und **UPDATE** möglich.

Für die Administration lassen sich zusätzlich die folgenden Berechtigungen vergeben:

- **FILE** – erlaubt das Lesen und Schreiben von Dateien auf dem Server
 - **GRANT** – erlaubt die Weitergabe der eigenen Berechtigungen an andere User
 - **PROCESS** – erlaubt den Abruf der aktuellen Prozessliste des Servers
 - **RELOAD** – ermöglicht den Aufruf der **FLUSH**-Anweisungen zum Einlesen geänderter Berechtigungen
 - **SHUTDOWN** – ermöglicht das Anhalten des MySQL-Servers
- Soll ein Benutzer sämtliche Berechtigungen erhalten, genügt es, das Recht **ALL** zu vergeben. Lediglich das **GRANT**-Recht muss hier noch explizit vergeben werden.

der Version 5 neu dazugekommen ist. Ganz neu sind die PHP Data Objekts (PDO), denen ein eigener Workshop in diesem Heft gewidmet ist (siehe Seite 82).

Der erste Schritt bei der Datenbankprogrammierung ist immer der Verbindungsaufbau zum Server. Als Parameter werden dabei Hostname, Benutzername und Passwort übergeben. Anschließend wird eine Verbindung zur Datenbank hergestellt. Bei der *mysqli*-Variante sind beide Schritte zusammengefasst, wie die folgenden Code-Beispiele zeigen:

```
//mysql-Variante
$verbindung = mysql_connect ($host,
$user, $pass);
$datenbank = mysql_select_db ($db);
//mysqli-Variante
$verbindung = new mysqli ($host, $user,
$pass, $db);
```

An eine so hergestellte Datenbankverbindung können nun beliebige SQL-Befehle übermittelt werden. Ein einfaches Beispiel zeigt das Prinzip:

```
// ausführen einer SQL Anfrage
$query = "SELECT * FROM Meine_Tabelle";
$result = mysql_query($query);

// Ausgabe der Ergebnisse in HTML
echo "<table>\n";
while ($line = mysql_fetch_array($result,
MYSQL_ASSOC)) {
    echo "\t<tr>\n";
    foreach ($line as $col_value) {
        echo "\t\t<td>$col_value</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";
```

```
// Freigeben des Resultsets und Schließen
der Verbindung
mysql_free_result($result);
mysql_close($link);
```

Die *mysqli*-Variante für die Übermittlung der SQL-Abfrage würde so aussehen:

```
$query = verbindung->query($sqlstring);
```

Da alle SQL-Befehle über dieses einheitliche Schema per PHP an den Datenbankserver abgesetzt werden, ist dies auch die entscheidende Funktion in Sachen MySQL-Programmierung. Auch die Manipulationen an den Datenbankinhalten erfolgt auf diese Weise.

Um die volle Leistungsfähigkeit von MySQL mit PHP ausschöpfen zu können, ist eine genaue Kenntnis der SQL-Abfragesprache unerlässlich. Die darüber hinaus in PHP zur Verfügung stehenden Funktionen für MySQL-Operationen sind in der PHP-Dokumentation genau beschrieben (php.net/manual/de/ref.mysql.php beziehungsweise php.net/manual/de/ref.mysqli.php).

»MySQL-Tuning«

Gerade bei dynamischen Webseiten sind Datenbankzugriffe der entscheidende Performance-Faktor. Die Optimierung von MySQL-Datenbanken und Datenbankzugriffen ist eine komplexe Angelegenheit, bei der es mehrere Faktoren zu berücksichtigen gilt. Zuerst geht es einmal darum, zu analysieren, wo im Gesamtsystem Flaschenhälse die Performance hemmen können.

Wie jede andere Software werden auch Datenbankoperationen nachhaltig von der Hardware beeinflusst, auf der sie ablaufen. Die Rechnung ist einfach: Je schneller einer Festplatte, desto höher ist auch die Geschwindigkeit bei Zugriffen auf eine MySQL-Datenbank. Auch die CPU und die Dimension des Hauptspeichers spielen bei Datenbankoperationen eine entscheidende Rolle.

Das Geschwindigkeitsverhalten im Live-Betrieb zu analysieren ist schwierig. Daher ist es sinnvoll, eine aufgesetzte Web-Applikation mit Hilfe von Benchmark- und Stress-Tools einer Belastungsprobe zu unterziehen. Wie beim Apache-Server gibt es auch für Datenbank-Server zahlreiche Tools, mit denen Sie diese Aufgabe erledigen können. MySQL (www.mysql.de) stellt mit der MySQL-Benchmark-Suite ebenfalls eine solche Testumgebung zur Verfügung.

Ein weiteres Tool, mit dem Sie einen Stress-Test durchführen können, ist Super Smack. Dieses und weitere Programme für Benchmarks und Stress-Tests finden Sie im Download-Bereich von www.mysql.com unter der Rubrik *Contributed Software*.

»Applikation optimieren«

Ein Großteil der Datenbank-Programmierung besteht in der Regel aus Anfragen an eine Datenbank per *select*. Hier gibt es auch reichlich Spielräume für Tuning und Optimierung. Ein wichtiger Punkt betrifft die Berechtigungen, die für den User gesetzt sind, der die Abfrage durchführt. Je komplexer das Berechtigungssystem ist, desto eher entsteht ein Zeit raubender Overhead. Denn können Sie dadurch vermeiden, dass Sie nur die Berechtigungen setzen, die das Skript auch benötigt. Reine Administrator-Rechte wie zum Beispiel zum Löschen von Datenbanken oder Tabellen werden in der Regel im Rahmen einer Web-Applikation nicht benötigt.

Mit dem Befehl *optimize table* können Sie eine MySQL-Tabelle optimieren und damit auch den Zugriff beschleunigen. Sie sollten diesen Befehl immer dann anwenden, wenn Sie große Teile der Tabelle gelöscht haben oder bei Tabellen mit Zeilen variabler Länge viele Änderungen durchgeführt haben. Das betrifft vor allem Tabellen, die *varchar*-, *blob*- oder *text*-Spalten enthalten.

»Query-Cache nutzen«

Ab Version 4.0.1 besitzt der MySQL-Server einen Query-Cache. Dort wird der Text einer *select*-Anfrage zusammen mit dem entsprechenden Ergebnis, das an den Client gesendet wird, gespeichert. Bei einer weiteren identischen Anfrage kann der Server die Ergebnisse aus dem Cache holen, statt dieselbe Anfrage noch einmal zu parsen und auszuführen.

Der Query-Cache kann seine Nützlichkeit vor allem in Umgebungen ausspielen, in denen sich Tabellen nicht

häufig ändern und in denen auf sie viele identische Anfragen erfolgen. Bei klassischen Content-Management-Systemen, bei denen die Inhalte nicht kontinuierlich aktualisiert und verändert werden, kann dies einen erheblichen Zuwachs an Speed mit sich bringen.

»Optimale Tabellenstruktur«

Wenn Sie Ihre Tabellenstruktur so anlegen, dass die Daten möglichst wenig Platz auf der Platte und im Arbeitsspeicher benötigen, kann das die Performance erheblich verbessern, weil Lesezugriffe von der Platte schneller ablaufen und weniger Hauptspeicher benötigt wird. Das Indexieren nimmt darüber hinaus weniger Ressourcen in Anspruch,

wenn es auf kleinere Spalten durchgeführt wird. MySQL unterstützt viele verschiedene Tabellentypen und Zeilenformate. Benutzen Sie möglichst die effizientesten Typen. Der Typ *mediumint* ist zum Beispiel besser als *int*, wenn es um Geschwindigkeit geht. Deklarieren Sie Spalten, wenn es möglich ist, immer als *not null*. Das macht alles schneller und Sie sparen ein Bit pro Spalte. Vermeiden Sie, einfach alle Spalten auf *null* zu haben, nur weil dies die Voreinstellung ist.

Wenn Sie keine Spalten variabler Länge benutzen (*varchar*, *text* oder *blob*), wird ein Festgrößenformat benutzt, das schneller ist.

Der primäre Index einer Tabelle sollte so kurz wie möglich sein. Das macht die Identifikation einer

Zeile sehr schnell. Kürzere Indexe sind nicht nur schneller, weil sie weniger Plattenplatz brauchen, sondern auch, weil Sie mehr Treffer im Index-Cache halten können und daher weniger Festplattenzugriffe benötigen.

Erzeugen Sie nur die Indexe, die Sie tatsächlich brauchen. Indexe sind zwar gut für schnelle Abfragen, aber schlecht beim Speichern von Daten.

Unter bestimmten Umständen kann es vorteilhaft sein, eine Tabelle zu teilen, die sehr oft gesamt wird. Das gilt insbesondere, wenn sie ein dynamisches Tabellenformat hat und Sie durch die Zerlegung kleinere Tabellen mit statischem Format erhalten.

»Weitere Optimierungstipps«

Benutzen Sie persistente Verbindungen zur Datenbank. Wenn das auf Grund der Systemgegebenheiten nicht geht, sollten Sie den Wert der *thread_cache_size*-Variablen im Konfigurations-File ändern. Überprüfen Sie immer, ob Ihre Anfragen auch tatsächlich die Indexe benutzen, die Sie in den Tabellen erzeugt haben. Sie können dazu den *explain*-

Laufzeit-Informationen

Dieser MySQL-Server läuft bereits 13 Tage, 12 Stunden, 22 Minuten und 43 Sekunden. Er wurde am 25. Februar 2004 um 20:38 gestartet.

- **Servertraffic:** In diesen Tabellen wird der Netzwerktraffic dieses MySQL-Servers seit dessen Start aufgeführt.

Traffic	# pro Stunde	Verbindungen	# pro Stunde	%
Empfangen	2.647 KB	Fehlgeschlagen	1	0,00
Gesendet	12.891 KB	Abgebrochen	0	0,00
Insgesamt	15.538 KB	Insgesamt	8.228	25,37
- **Abfragestatistik:** Seit seinem Start wurden 41.331 Abfragen an diesen MySQL-Server gesandt.

Abfrageart	# pro Stunde	%	Abfrage
admin commands	0	0,00	repair
alter table	2	0,01	replace
analyze	0	0,00	replace select
backup table	0	0,00	reset
begin	0	0,00	restore table
change db	5.492	16,83	revoke
change master	0	0,00	rollback
check	0	0,00	select
commit	0	0,00	set option
create db	1	0,00	show binlogs
create function	0	0,00	show create

The screenshot shows a PHP web page with a navigation menu on the left (PHP Handbuch, Funktionsreferenz, Apache, APC, Arrays, Appel, BC math, bootstrap, Soap2, Kalender, CVS, Classical, Klassen/Objekte, COM, ClibPDF, Cmsk, etags, GURL, Cybercash, CyberMUT, Cymid IMAP, Datum/Zeit, Bus, Bbase, DSM, DE++, db, Direkte Ein-/Ausgabe, Verzeichnisse, SOAP). The main content area is titled 'LXXXIII. Verbesserte MySQL Erweiterung' and includes sections for 'Einführung', 'Warnung', 'Anforderungen', and 'Installation'. The 'Warnung' section states: 'Diese Erweiterung ist EXPERIMENTELL. Das Verhalten dieser Erweiterung, einschließlich der Funktionsnamen, und alles Andere was hier dokumentiert ist, kann sich in zukünftigen PHP-Versionen ohne Ankündigung ändern. Seien Sie gewarnt und verwenden Sie diese Erweiterung auf eigenes Risiko.'

Befehl benutzen. Versuchen Sie, komplexe *select*-Anfragen auf Tabellen zu vermeiden, die viel aktualisiert werden.

Sortieren Sie eine Tabelle komplett neu in der Reihenfolge, in der Sie die Daten im Skript am häufigsten benutzen. Dies kann nach größeren Änderungen einen erheblichen Geschwindigkeitsvorteil mit sich bringen.

Normalerweise nützt es nichts, eine Datenbanktabelle in verschiedene Tabellen aufzuteilen. Der einzige Fall, wo es wirklich etwas ausmacht, ist, wenn die Tabelle dynamische Zeilenlänge hat, was nicht in eine feste Zeilenlänge umgewandelt werden kann, oder wenn Sie die Tabelle sehr oft scannen müssen, die meisten der Spalten hierfür aber nicht benötigen.

Wenn Sie sehr oft etwas auf der Grundlage von Informationen aus sehr vielen Zeilen berechnen müssen, ist es in den meisten Fällen besser, eine neue Tabelle einzuführen und den Zähler in Echtzeit zu aktualisieren.

In einigen Fällen bringt es mehr Speed, Daten zu komprimieren und in einem Blob zu speichern. In diesem Fall müssen Sie in Ihrer Applikation zusätzlichen Code einbauen, um die Daten im Blob zu packen beziehungsweise zu entpacken. Das kann aber unter Umständen etliche Zugriffe einsparen und die Performance erhöhen.

Normalerweise sollten Sie versuchen, Daten nicht redundant zu speichern. Scheuen Sie sich aber nicht, von dieser Prämisse abzuweichen, wenn Sie dadurch einen bedeutenden Geschwindigkeitsvorteil erzielen können.

Benutzen Sie *insert /*! delayed */*, wenn Sie nicht wissen müssen, wann Ihre Daten geschrieben werden. Das erhöht die Geschwindigkeit, weil viele Datensätze mit einem einzigen Festplatten-schreibzugriff geschrieben werden können.

Benutzen Sie *insert /*! low_priority */*, wenn Sie wollen, dass Ihre Selects höhere Priorität haben.

Mit *select /*! high_priority */* können Sie *selects* in der Warteschlange nach vorn bringen. Ein solcher *select* wird sogar dann durchgeführt, wenn jemand darauf wartet, etwas zu schreiben.

Benutzen Sie das mehrzeilige *insert*-Statement, um viele Zeilen mit einem SQL-Befehl zu speichern.

Bei einer normalen Webserver-Konfiguration sollten Bilder als separate Dateien gespeichert werden. Das heißt, Sie speichern nur einen Verweis zur Datei in der Datenbank. Normale Webserver können Dateien viel besser cachen als Datenbankinhalte. Daher ist dies ein sehr einfacher Weg zu mehr Speed.

Spalten mit identischen Informationen in unterschiedlichen Tabellen sollten identisch deklariert sein und identische Namen haben. Versuchen Sie, die Namen kurz zu halten.

»Security«

Ein wichtiger Punkt bei Web-Applikationen mit einer PHP-MySQL-Kombination ist die Sicherheit. Von Haus aus bietet der MySQL-Server eine Fülle unterschiedlicher Sicherheitsfeatures, die einem geschulten Administrator eine Vielzahl von Schutzmöglichkeiten für die sensiblen Daten zur Verfügung stellen. Zentraler Ansatzpunkt aller Sicherheitsüberlegungen in Sachen MySQL ist das Benutzerkonzept der Datenbank.

Ähnlich wie auf einem Linux-System besitzt der User *root* auch für den MySQL-Server uneingeschränkte Zugriffsrechte. Dies birgt einige Risiken, wenn Sie diesen User in irgendwelche Applikationen einbinden. Für die Entwicklung einer Web-Applikation, die Zugriff auf die Datenbank benötigt, sollten Sie immer weitere Benutzer einrichten, deren Berechtigungen auf einzelne Datenbanken oder Tabellen sowie den Zugriff von bestimmten IP-Adressen beschränkt sind. MySQL bietet sogar die Möglichkeit, die Berechtigungen auf Spalten einzelner Tabellen zu beschränken. Für

die Vergabe und den Entzug von Berechtigungen auf dem MySQL-Server sorgen die Befehle *GRANT* und *REVOKE*.

Der Kasten »MySQL-Benutzerrechte« auf Seite 28 gibt eine Übersicht zu den verschiedenen Berechtigungen, die in der Benutzerverwaltung zur Verfügung stehen. Das Einrichten und Verwalten der Benutzer erfolgt entweder über den MySQL-Client, der von der Konsole beziehungsweise Shell aus aufgerufen wird, oder alternativ über Admin-Tools wie das erwähnte PhpMyAdmin. Der Kasten »MySQL-Benutzerverwaltung« auf Seite 27 zeigt, in welchen Tabellen MySQL die Rechte verwaltet.

Neben den Benutzern des MySQL-Servers sollte man auf einem Linux-System auch die Berechtigungen im Auge behalten, mit denen der MySQL-Server-Prozess arbeitet und mit denen die Dateien im Filesystem abgelegt sind. Der MySQL-Server sollte niemals als Benutzer *root* laufen, sondern über ein eigenes Benutzerkonto verfügen, das keine Administrationsrechte besitzt. Weiterhin empfiehlt sich ein Blick auf das MySQL-Datenverzeichnis. Hier legt MySQL den kompletten Inhalt aller Tabellen ab. Sämtliche Dateien sollten vor dem Zugriff durch Unberechtigte geschützt sein, um so den Datendiebstahl zu verhindern.

»Sicher programmieren«

Die Sicherheit des MySQL-Servers obliegt nicht nur dem Administrator allein. Jede einzelne Anwendung, die mit dem MySQL-Server Daten austauscht, muss auf die Sicherheit und Korrektheit der übertragenen Informationen achten. Um vor böswilligen Manipulationen bei Formulareingaben der Web-Applikation geschützt zu sein, sollte jede Eingabe vor der Übergabe an die Datenbank geprüft werden. In PHP sollte die Funktion *mysql_escape_string()* genutzt werden, um einen Eingabe-String für die Verarbeitung in MySQL korrekt zu maskieren. [mb]

MySQL-Datenbanken administrieren

Ob Shared Hosting oder dedizierter Server – ohne ein leistungsfähiges Administrations-Tool zur Verwaltung der Datenbanken geht es nicht. Einen hervorragenden Ruf in diesem Bereich genießt das Tool PhpMyAdmin.

»Funktionen von PhpMyAdmin«

Das Tool bietet eine umfassende Funktions-Sammlung zum Administrieren einer MySQL-Datenbank. Schon auf der Eingangsseite bietet es zum Anlegen einer Datenbank ein Eingabefeld an, in das man lediglich den Namen der Datenbank eintragen muss. Schon ist die neue Datenbank erstellt. Shared-Hosting-Mieter haben von dieser Funktion jedoch vielfach nichts, da in diesem Bereich vordefinierte Datenbanknamen die Regel sind.

Das Anlegen von Tabellen ist jedoch in jeder Installationsumgebung möglich. In der Datenbankansicht bietet PhpMyAdmin die Möglichkeit, neue Tabellen zu definieren. Dazu muss in das Feld *Name* der gewünschte Tabellenbezeichner und in *Felder* die Anzahl der gewünschten Spalten eingetragen werden. Nach diesem Schritt erfolgt die De-

finition der einzelnen Spalten. Für jede zu definierende Spalte wird dem HTML-Formular eine Zeile hinzugefügt. Dabei müssen in mehreren Feldern die Eigenschaften der einzelnen Spalten festgelegt werden. Als Erstes sollte immer ein Primärschlüssel der Tabelle definiert werden, damit jeder Datensatz eindeutig identifiziert werden kann.

Nach dem Anlegen der Tabelle wechselt PhpMyAdmin in die entsprechende Tabellenansicht. Oben wird das SQL-Kommando angezeigt, mit dem die Tabelle erzeugt wurde. Auf der Übersichtsseite haben Sie nun die Möglichkeit, neue Spalten hinzuzufügen, bestehende zu modifizieren oder gar zu löschen, neue Indizes zu definieren, Datensätze einzufügen oder die Tabelle umzubenennen.

Hervorragende Dienste leistet PhpMyAdmin bei der Installation fertiger PHP-Skripts. Bei größeren Applikationen mit Datenbankbindung stehen die notwendigen Tabellendefinitionen als so genannter SQL-Dump zur Verfügung. Es handelt sich dabei um eine Batch-Datei mit allen notwendigen SQL-Befehlen. PhpMyAdmin bietet nun die Mög-

Hat man die Auswahl für ein System und einen Provider getroffen, stellt sich die Frage, wie man die verfügbare Datenbankumgebung am besten verwaltet und administriert. Auch bei einem Shared-Hosting-Paket, bei dem die Eingriffsmöglichkeiten systembedingt eingeschränkt sind, lohnt es sich, sich mit einem passenden Tool zum Anlegen von Tabellen, dem komfortablen Import und Export von Daten und für Backup-Aufgaben zu bewaffnen.

Bei einer Server-Umgebung ist dies geradezu ein Muss, es sei denn, man beherrscht die MySQL-Administration auf Kommandozeilen-Ebene perfekt. Bei den Administrations-Tools kann man zwei Varianten unterscheiden: solche, die auf dem Server oder dem Webservice installiert werden und solche, die als Client auf dem lokalen PC eingerichtet und betrieben werden. Wie erwähnt, funktioniert diese Methode bei Shared-Hosting-Umgebungen jedoch in den wenigsten Fällen.

Die Hitparade der Tools wird angeführt von der Applikation PhpMyAdmin (www.phpmyadmin.net). Sie ist vollständig in PHP programmiert und bietet einen breiten Funktionsumfang. Da eine Installation von PhpMyAdmin im Document-Root eines Web-Auftritts nicht ohne Risiko ist, bieten viele Webhoster PhpMyAdmin bei ihren Shared-Hosting-Paketen bereits vorinstalliert an. Wo dies nicht der Fall ist, sollte man PhpMyAdmin in jedem Fall entweder außerhalb des Webverzeichnis-Baums oder – wenn dies nicht möglich ist – in einem geschützten Unterverzeichnis installieren. Zumindest die zweite Variante ist auch bei Shared-Hosting-Paketen jederzeit möglich.



Für die Bearbeitung von Tabellen bietet PhpMyAdmin eine komfortable grafische Oberfläche an.

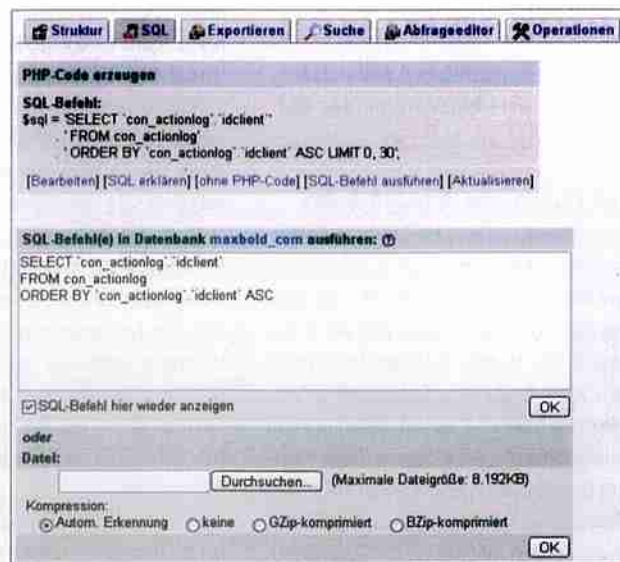


Auch die MySQL-Benutzerverwaltung lässt sich über geeignete Admin-Tools komfortabel abwickeln.

PostgreSQL ist im Webbereich eine attraktive Datenbank-Alternative zu MySQL.



Auch PHP-Code aus interaktiven SQL-Abfragen kann man mit Hilfe von PhpMyAdmin erzeugen.



lichkeit, per Upload diese Datei einzulesen und die Befehle ausführen zu lassen. Damit lassen sich auch komplexe Tabellenkonstrukte, wie sie zum Beispiel für Foren, Weblogs oder Content-Management-Systeme erforderlich sind, in wenigen Minuten realisieren.

»Inhalte sichern«

In Sachen Datenbank-Backup leistet PhpMyAdmin ebenfalls wertvolle Dienste. Grundsätzlich bietet MySQL zum Sichern von Datenbanken und Tabellen drei Möglichkeiten an: das Kopieren der Datenbankverzeichnisse und der enthaltenen Dateien, die Verwendung von speziellen SQL-Kommandos sowie den Export der Daten in ein universelles Format.

Ein Kopieren der Datenbankverzeichnisse inklusive der Datenbanken und Strukturen ist nur auf einem Root-Server möglich. Shared-Hosting-User können ihre Backups bequem mit den Export-Funktionen von PhpMyAdmin durch-

führen und dabei auch noch unterschiedliche Ausgabeformate wählen.

Klicken Sie für eine Backup-Operation in der Tabellenansicht von PhpMyAdmin auf den Reiter *Exportieren*. Als Formate für den Export eignen sich SQL, CSV-Daten und XML am besten. Im SQL-Format exportiert PhpMyAdmin die Daten als SQL-Kommandodatei. Der große Vorteil ist bei dieser Variante, dass dabei alle Informationen zu der Datenbank und den Tabellen sowie die Daten in ausführbare SQL-Kommandos umgewandelt werden. Ist irgendwann einmal ein Rückspielen der Daten notwendig, müssen Sie diese Datei lediglich ausführen, und die Datenbank, die Tabelle sowie die Datensätze werden wiederhergestellt.

Bei Verwendungen des CSV-Formats werden lediglich die Datensätze gesichert. Für jeden Datensatz wird in der CSV-Datei eine neue Zeile eingefügt und die einzelnen Felder der Datensätze werden mit Strichpunkten (Semikolons) voneinander getrennt. Leider geht hierbei die Struktur der Datenbank verloren.

Das XML-Format entspricht im Wesentlichen dem CSV-Format. Statt die Daten aber einfach in Zeilen und durch Semikolons zu trennen, wird die Hierarchie beibehalten und die Daten werden in Form von Tags strukturiert. Zwar geht so die Struktur der Datenbank nicht verloren, doch es fehlen wichtige Informationen zum Tabellenaufbau.

Je nach Größe der Tabelle ist es sinnvoll, die Daten zu komprimieren. Wählen Sie unter *Kompression* dazu eine der Optionen *keine*, *Zip-komprimiert* (für Windows-Nutzer zu empfehlen) oder *GZip-komprimiert*.

»Daten importieren«

Auch die Import-Funktion von PhpMyAdmin leistet wertvolle Dienste. Einmal kann man natürlich auf diese Weise eine gesicherte Tabellenkonstruk-

tion im Fall eines Crash wieder in das Produkktivsystem zurückspielen. Jedoch kann man mit diesem Verfahren jederzeit auch größere Datenmengen in einem Fremdformat in eine vorhandene Tabelle einspielen, ohne mühsam Formulare ausfüllen zu müssen. Für den Import stehen dieselben Formate zur Verfügung wie für den Export, also SQL, CSV oder XML.

Beim Einspielen von CSV-Daten kann man den Input über Parameter sehr genau spezifizieren. In der Regel sind die einzelnen Felder eines Datensatzes durch Semikolons und Datensätze durch Zeilenumbrüche voneinander getrennt. Wenn Sie eine Datei mit Feldern, die durch Tabulatoren getrennt sind, importieren, tragen Sie in das Feld *Felder getrennt mit* einfach `\t` ein. Sollte die CSV-Datei nicht durch die unter Windows übliche Zeichenfolge `\r\n` (Wagenrücklauf und Zeilenumbruch), sondern nur durch die unter Linux typische Zeichenfolge `\n` (Zeilenumbruch) getrennt sein, dann passen Sie dies entsprechend in dem Feld *Zeilen getrennt mit* an, zum Beispiel durch die Eingabe von `\n`.

Finden sich in der zu importierenden Datei nicht zu jeder Spalte der MySQL-Tabelle entsprechende Felder in den neuen Datensätzen, dann können Sie die Reihenfolge und Zuordnung in dem Feld *Spaltennamen* festlegen.

»SQL-Anweisungen«

Außerordentlich hilfreich – insbesondere für PHP-Programmierer – ist die Möglichkeit, direkt SQL-Anweisungen mit Hilfe von PhpMyAdmin auszuführen. Damit lassen sich Befehle auf einfache Weise vor dem Codieren kurz testen. Auch beim Finden von Fehlern in einem Skript, in dem SQL-Anweisungen verarbeitet werden, kann diese Funktion nützlich Dienste leisten. Der von PhpMyAdmin generierte PHP-Code lässt sich per Cut and Paste einfach und schnell in ein Skript übertragen.

Wenn Sie sich in der Tabellenansicht befinden, können Sie weiter unten auf der Seite ein großes Text- und eine kleines Auswahlfeld entdecken. In das Textfeld können Sie eine beliebige SQL-Anweisung eingeben. Das daneben befindliche Auswahlfeld zeigt die Felder der gerade ausgewählten Tabellen an. Mit einem Klick auf *OK* wird die Anweisung ausgeführt und das Ergebnis ausgegeben. Sie können jedoch auch in dem Feld darunter eine SQL-Kommandodatei, die Sie zum Beispiel auf Ihrem lokalen PC vorbereitet haben, auswählen und diese ausführen lassen.

Alternativ haben Sie die Möglichkeit, auf das kleine Icon mit der Aufschrift *SQL* zu klicken. Es öffnet sich daraufhin ein neues Fenster, in dem Sie sowohl SQL-Befehle direkt eintippen, eine Kommandodatei ausführen oder sich die bisherigen SQL-Kommandos anzeigen lassen können. Sollten Sie im Hauptfenster eine bestimmte Tabelle ausgewählt haben, dann sind die entsprechenden Spalten dieser Tabelle ebenfalls in dem Befehlsfenster verfügbar. Das Ergebnis der Befehle wird im Hauptfenster ausgegeben.

Ist es Ihnen hingegen lieber, eine SQL-Anweisung mittels *Point and Click* zusammenzustellen anstatt sie einzutippen, so wählen Sie auf der linken Seite einfach eine beliebige Datenbank aus und klicken Sie anschließend auf den Reiter *Abfrageeditor*. Dieser bietet Ihnen nun die Option, die gewünschten Felder für die Abfrage auszuwählen, Kriterien und Verknüpfungen festzulegen und diese Abfrage anschließend auszuführen.

PhpMyAdmin ist ein komfortables und mächtiges Werkzeug zur Verwaltung von MySQL-Datenbanken, auf das auch Shared-Hosting-Mieter nicht verzichten sollten. Wenn der Provider das Tool noch nicht installiert hat, sollten Sie das nachholen – allerdings aus Sicherheitsgründen unbedingt in einem geschützten Verzeichnis. [mb]

Info

Alternative MySQL-Tools

Komfortable grafische Bedienoberflächen statt fehlerträchtiger Kommandozeilen: Zahlreiche Skripts und Programme erleichtern die Arbeit mit Datenbanken erheblich.

ADODB

Da in PHP alle Datenbanken unterschiedlich angesprochen werden, schaltet sich ADODB als Vermittler zwischen PHP-Skript und Datenbank ein und sorgt für eine einheitliche Schnittstelle, über die mit allen Datenbanken kommuniziert werden kann. ADODB unterstützt derzeit mehr als ein Dutzend verbreiteter Datenbanken: MySQL, PostgreSQL, Interbase, Firebird, Informix, Oracle, MS SQL 7, Foxpro, Access, ADO, Sybase, Frontbase, DB2 und Generic ODBC. Die Installation von ADODB ist denkbar einfach. Damit ADODB funktioniert, muss auf dem Webserver mindestens PHP 4.05 oder höher installiert sein. Dank einfacher Funktionen sind kleine PHP-Skripts für MySQL oder andere Datenbanken im Handumdrehen ADODB-kompatibel. Der Verbindungsaufbau ist zwar etwas länger als für MySQL, aber dafür ebenso logisch aufgebaut. adodb.sourceforge.net

CFMyAdmin – MySQL Database Manager

Ein MySQL-Datenbank-Manager auf der Basis von Cold Fusion, der sich im Funktionsumfang mit dem PHP-Äquivalent PhpMyAdmin messen kann, ist CFMyAdmin. Datenbanken und Tabellen lassen sich damit in einer Windows-ähnlichen Umgebung erstellen und pflegen. Das Skript unterstützt mehrere Datenbankverbindungen gleichzeitig und bietet auch diverse Import- und Export-Funktionen. Für die Arbeit mit SQL-Kommandos steht ein eigener SQL-Editor zur Verfügung. Für die Arbeit mit mehreren Datenbanken können Profile angelegt und abgespeichert werden. Die Installation erfolgt über ein Setup-Programm. www.cfmyadmin.com

Dadabik

Die Applikation Dadabik ist ein universelles Interface zur Pflege von MySQL-Datenbanken. Damit lassen sich schnell und einfach Datensätze hinzufügen, ändern oder löschen. Es sind nur wenige Parameter in der Datei *include/config.php* zu ändern. Insbesondere die Zugangsparameter zur MySQL-Datenbank sind hier einzutragen. Dann werden alle Files des Ordners *program_files* aus dem Archiv in ein Verzeichnis auf den Server übertragen, und schon es kann losgehen. Der Aufruf des Skripts *install.php* startet die Installation von Dadabik, mit *admin.php* können individuelle Anpassungen vorgenommen werden. Der eigentliche Programmaufruf zum Bearbeiten einer MySQL-Datenbank erfolgt über *index.php*. www.dadabik.org

Matt's MySQL Control Panel

Ein PHP-Skript, mit dessen Hilfe MySQL-Datenbanken gepflegt werden können. Außerdem lassen sich damit Dumps importieren und Daten aus einer Tabelle exportieren. Das Skript bietet für die Oberfläche diverse Sprachoptionen, bei denen Deutsch allerdings fehlt. Über ein Konfigurations-File können einige Voreinstellungen vorgenommen werden. Die Installation ist einfach: Alle Files des Archivs müssen nur auf den Server kopiert werden, dann kann man sofort loslegen. www.mattsscripts.co.uk/mpanel.htm

MySQL Commander

Der MySQL Commander erlaubt es, schnell und einfach Backups von Datenbanken auf einem MySQL-Server zu erstellen. Die Daten werden in Textdateien gespeichert und können jederzeit wieder in die Datenbank zurückgeschrieben

werden. Neben den Inhalten lässt sich auch die Struktur der Tabellen sichern.

www.bitesser.de

MySQL Datamanagement

MySQL Datamanagement ist ein Tool, um die Daten einer Datenbank zu verwalten. Alle Tabellen der Datenbank werden angezeigt; der Anwender kann die üblichen Manipulationen an den Datensätzen vornehmen wie Auflisten, Sortieren, Editieren, Löschen und Hinzufügen. sourceforge.net/projects/datamanagement

Phormation

Phormation ist ein Satz von PHP-Funktionen, die bei der Entwicklung datenbankgestützter HTML-Formulare helfen. Unterstützt werden PostgreSQL, MySQL und ODBC. Um Webseiten schneller zu entwickeln, enthält Phormation wiederverwendbare Widgets, das sind Grafikobjekte einer Bedienoberfläche – beispielsweise Select-Boxen, Buttons, Textfelder oder ein Datumfeld. www.peaceworks.ca/phormation.php

MySQL Administrator

MySQL Administrator ist eine leistungsfähige grafische Verwaltungskonsolle, über die Sie Ihre MySQL-Umgebung verwalten können. Das Tool integriert Wartung und Administration einer Datenbank in eine einzige, einheitliche Umgebung mit klar strukturierter und intuitiv bedienbarer Bedienoberfläche. MySQL Administrator ermöglicht Entwicklern und Datenbankadministratoren, alle Operationen der Befehlszeilenebene über die grafische Bedienoberfläche auszuführen – Server konfigurieren, Benutzer verwalten und Zustände der Datenbanken überwachen. Auch andere Arbeiten wie die Überwachung von Datenbankreplikationen, Backup und Restore und das Überprüfen von Protokolldateien können über die grafische Bedienoberfläche von MySQL Administrator erledigt werden. MySQL Administrator ist für Microsoft Windows und Linux verfügbar und kompatibel mit MySQL 4.x. www.mysql.de/products/administrator