

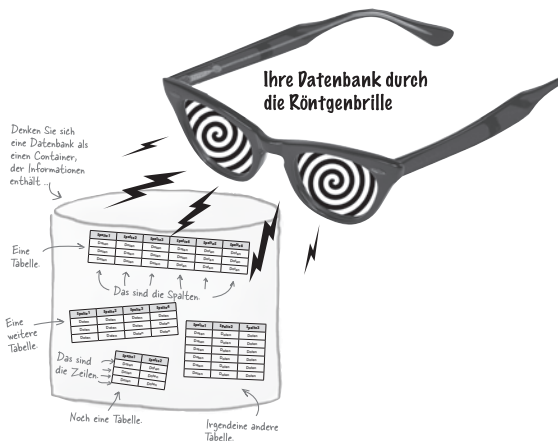
Daten und Tabellen

1

Ein Ort für alle Dinge

Hassen Sie es auch, Dinge zu verlieren? Egal ob es Ihre Autoschlüssel, der 25%-Gutschein von Globetrotter oder Ihre Anwendungsdaten sind: Es gibt nichts Schlimmeres, als nicht dazu in der Lage zu sein, **das, was man braucht, zur Hand zu haben** ... wenn man es braucht. Und wenn es um Ihre Anwendungen geht, gibt es keinen besseren Ort, Ihre wichtigen Informationen zu speichern, als eine **Tabelle**. Blättern Sie also um, treten Sie ein und machen Sie einen Spaziergang durch die Welt der **relationalen Datenbanken**.

Daten definieren	2
Betrachten Sie Daten in Kategorien	7
Was in einer Datenbank steckt	8
Ihre Datenbank durch die Röntgenbrille	10
Datenbanken enthalten verwandte Daten	12
Tabellen unter der Lupe	13
Übernehmen Sie das Kommando!	17
Die Tabelle einrichten: die CREATE TABLE-Anweisung	19
Eine kompliziertere Tabelle erstellen	20
Wie einfach das Schreiben von SQL doch ist	21
Endlich die Tabelle meine_kontakte erstellen	22
Ihre Tabelle ist fertig	23
Machen Sie Bekanntschaft mit ein paar Datentypen	24
Eine Beschreibung Ihrer Tabelle	28
Eine Datenbank oder Tabelle kann nicht neu erstellt werden!	30
Raus mit der alten Tabelle, rein mit der neuen	32
Daten einfügen mit der INSERT-Anweisung	34
Die INSERT-Anweisung erstellen	37
Variationen zu einer INSERT-Anweisung	41
Spalten ohne Werte	42
Blicken Sie mit der SELECT-Anweisung in Ihre Tabelle	43
SQL im Gespräch: Geständnis einer NULL	44
Ihre innere NULL beherrschen	45
NOT NULL erscheint in DESC	47
Die Lücken mit DEFAULT füllen	48
Ihr SQL-Werkzeugkasten	50



Die *SELECT*-Anweisung**Geschenker Datenabruf**

2

Ist Geben wirklich besser denn Nehmen? Bei Datenbanken ist es nicht unwahrscheinlich, dass Sie genauso oft **Daten abrufen** wie einfügen müssen. An dieser Stelle betritt dieses Kapitel die Bühne: Sie treffen die mächtige **SELECT**-Anweisung und lernen, wie man **Zugang zu den wichtigen Informationen erhält**, die Sie in Ihre Tabellen gesteckt haben. Sie werden sogar lernen, wie man mit **WHERE**, **AND** und **OR** Daten selektiv abrufen und so vermeidet, Daten anzuzeigen, die man *nicht* braucht.

Ein Date oder kein Date?	54
Ein besseres SELECT	57
Was ist dieses *?	58
Wie Sie Ihre Datentypen abfragen	64
Mehr Interpunktionsprobleme	65
Ungebundene Anführungszeichen	66
Einfache Anführungszeichen sind Sonderzeichen	67
Daten mit einfachen Anführungszeichen einfügen	68
Bestimmte Spalten auswählen, um das Ergebnis einzugrenzen	73
Schnellere Ergebnisse mit ausgewählten Spalten	73
Ihre Abfragen kombinieren	80
Numerische Werte finden	83
Vergleichsoperatoren	86
Mit Vergleichsoperatoren numerische Daten finden	88
Mit Vergleichsoperatoren Textdaten einfangen	91
Sein ODER nicht sein	93
Der Unterschied zwischen AND und OR	96
NULLen finden mit IS NULL	99
Zeit sparen mit einem Schlüsselwort: LIKE	101
Der Joker sticht	101
Bereiche wählen mit AND und Vergleichsoperatoren	105
Aber es gibt einen besseren Weg	106
Nach dem Date ist man entweder IN ...	109
... oder NOT IN	110
Mehr NOT	111
Ihr SQL-Werkzeugkasten	116



DELETE und UPDATE

Veränderung tut gut

3

Sie ändern häufig Ihre Meinung? Das ist ab jetzt in

Ordnung! Mit den Befehlen, die Sie im Folgenden lernen werden – **DELETE** und **UPDATE** –, sind Sie nicht mehr an Entscheidungen gebunden, die Sie vor sechs Monaten gefällt haben, als Sie die Daten darüber eingefügt hatten, dass die Dauerwelle demnächst wieder in sei. Mit UPDATE können Sie **Daten ändern**, und mit DELETE werden Sie **Daten los**, die Sie nicht mehr brauchen. Aber wir geben Ihnen nicht nur die Werkzeuge. In diesem Kapitel lernen Sie auch, wie Sie gezielt mit Ihren neuen Kräften umgehen und vermeiden, Daten zu vernichten, die Sie noch brauchen.

Clowns sind gruselig	120
Clowns festhalten	121
Die Clowns sind unterwegs	122
Wie Ihre Clown-Daten eingefügt werden	126
Bonzo, wir haben ein Problem	128
Mit DELETE einen Datensatz loswerden	129
Unsere neue DELETE-Anweisung verwenden	131
DELETE-Regeln	132
Der INSERT-DELETE-Doppelschritt	135
Passen Sie bei Ihren DELETEDs auf	140
Der Ärger mit ungenauen DELETEDs	144
Daten ändern mit UPDATE	146
UPDATE-Regeln	147
UPDATE ist das neue INSERT/DELETE	148
UPDATE im Einsatz	149
Die Bewegungen der Clowns aktualisieren	150
Aktualisieren Sie Ihre Preise	154
Wir brauchen nur ein UPDATE	156
Ihr SQL-Werkzeugkasten	158



Ängstigen wir dich?

Cleveres Tabellen-Design

Warum normal sein?

4

Sie haben Tabellen erzeugt, ohne ihnen viele Gedanken zu schenken. Das war in Ordnung. Sie haben funktioniert. Sie können auf ihnen `SELECTs`, `INSERTs`, `DELETES` und `UPDATES` ausführen. Aber wenn Ihre **Datenmenge wächst**, beginnen Sie, **auf Dinge zu stoßen, bei denen Sie sich wünschten, Sie hätte sie anders gemacht**, damit jetzt Ihre `WHERE`-Klauseln einfacher wären. Und dazu müssen Sie Ihre Tabellen *normal* machen.

Zwei fischige Tabellen	160
Bei einer Tabelle geht es um Beziehungen	164
Atomare Daten	168
Atomare Daten und Ihre Tabellen	170
Regeln für atomare Daten	171
Gründe, normal zu sein	174
Die Vorteile normalisierter Tabellen	175
Clowns sind nicht normal	176
Auf halbem Weg zu 1NF	177
PRIMÄRSCHLÜSSEL-Regeln	178
Sich der NORMALITÄT nähern	181
Gregors Tabelle in Ordnung bringen	182
Unsere alte <code>CREATE TABLE</code> -Anweisung	183
Zeig mir die Table ^{Tabelle}	184
Zeitsparende Anweisung	185
<code>CREATE TABLE</code> mit einem <code>PRIMARY KEY</code>	186
1, 2, 3 ... und alles automatisch	188
Einer Tabelle einen <code>PRIMARY KEY</code> hinzufügen	192
<code>ALTER TABLE</code> und ein neuer <code>PRIMARY KEY</code>	193
Ihr <code>SQL</code> -Werkzeugkasten	194

Einen Moment. Ich habe eine Tabelle, die voller Daten ist. Sie können nicht ernsthaft erwarten, dass ich wie in Kapitel 1 den Befehl `DROP TABLE` einsetze und das ganze Zeug dann neu eintippe, nur um einen Primärschlüssel für jeden Datensatz zu erzeugen ...



ALTER

5

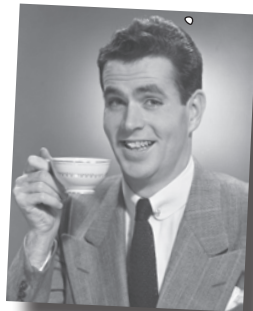
Die Vergangenheit umschreiben

Haben Sie sich jemals gewünscht, die Fehler der Vergangenheit korrigieren zu können? Jetzt haben Sie Gelegenheit dazu. Mit der **ALTER-Anweisung** können Sie all die Lektionen, die Sie gelernt haben, auf Tabellen zur Anwendung bringen, die Sie vor Tagen, Monaten oder sogar Jahren entworfen haben. Besser noch: Sie können es sogar tun, ohne dass das Einfluss auf Ihre Daten hat. Wenn Sie mit dem Folgenden durch sind, wissen Sie, was **normal** wirklich bedeutet, und können es auf alle Ihre Tabellen, alte und neue, anwenden.

Tabellen-Tuning

Wir müssen Änderungen vornehmen	198
Tabellenveränderungen	203
Extrem-Tabellenumbau	204
Die Tabelle umbenennen	205
Wir müssen einige Pläne machen	207
Ihre Spalten umfunktionieren	208
Strukturelle Veränderungen	209
ALTER und CHANGE	210
Zwei Spalten mit nur einer SQL-Anweisung ändern	211
Schnell! Löschen Sie diese Spalte	215
Ein genauer Blick auf die nichtatomare Spalte wohnort	222
Nach Mustern suchen	223
Ein paar praktische String-Funktionen	224
Eine neue Spalte mit einer aktuellen füllen	229
Wie unsere UPDATE/SET-Kombo funktioniert	230
Ihr SQL-Werkzeugkasten	232

Es ist an der Zeit, Ihre müde, alte Gebrauchtwagen-Tabelle in einen Kunden-Magneten zu verwandeln und sie auf ein Tabellenniveau zu heben, von dem Sie nicht einmal geträumt haben.



Fortgeschrittenes `SELECT`

6

Ihre Daten mit neuen Augen sehen

Es ist an der Zeit, Ihrem Werkzeugkasten etwas Finesse hinzuzufügen. Sie wissen bereits, wie man mit `SELECT` Daten auswählt und `WHERE`-Klauseln einsetzt. Aber manchmal benötigt man mehr **Genauigkeit**, als sie `SELECT` und `WHERE` bieten. In diesem Kapitel werden Sie lernen, wie Sie Ihre Daten **ordnen und gruppieren** und wie Sie auf Ihren Ergebnissen **mathematische Operationen durchführen**.

Die Videothek Datenhausen baut um	236
Probleme mit unseren aktuellen Tabellen	237
Die bestehenden Daten zuordnen	238
Die neue Spalte füllen	239
<code>UPDATE</code> mit einem <code>CASE</code> -Ausdruck	242
Es scheint, als hätten wir ein Problem	244
Tabellen können unordentlich werden	249
Die ausgewählten Daten sortieren	250
Probieren Sie ein kleines <code>ORDER BY</code>	253
Eine einzelne Spalte ordnen	254
<code>ORDER</code> mit zwei Spalten	257
<code>ORDER</code> mit mehreren Spalten	258
Eine sortierte <code>film_tabelle</code>	259
Die Reihenfolge mit <code>DESC</code> umkehren	261
Das Pfadfinderinnen-Keksverkaufsproblem	263
<code>SUM</code> kann für uns addieren	265
Mit <code>GROUP BY</code> alle auf einmal addieren	266
<code>AVG</code> mit <code>GROUP BY</code>	267
<code>MIN</code> und <code>MAX</code>	268
Mit <code>COUNT</code> die Tage zählen	269
<code>SELECT DISTINCT</code>	271
Die Anzahl an Ergebnissen limitieren	274
Auf nur den zweiten Platz eingrenzen	275
Ihr <code>SQL</code> -Werkzeugkasten	278

**VIDEOTHEK
DATENHAUSEN**

Eine Datenbank, viele Tabellen

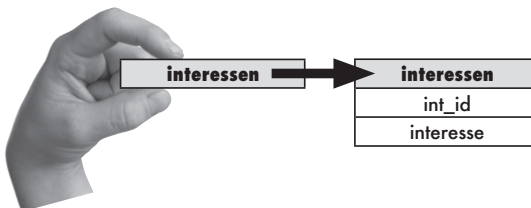
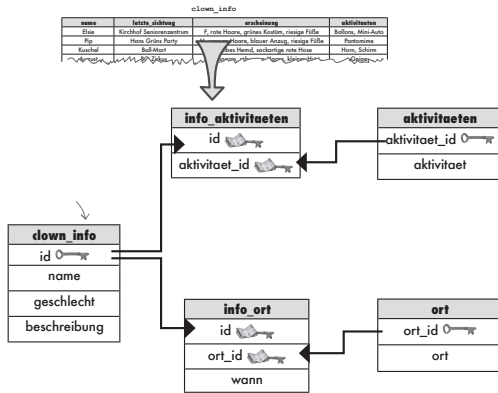
Aus Ihrer Tabelle herauswachsen

7

Manchmal reicht eine einzige Tabelle einfach nicht mehr

aus. Ihre Daten sind komplexer geworden, und die **eine Tabelle**, mit der Sie gearbeitet haben, **schafft es nicht mehr**. Ihre Tabelle steckt voller redundanter Daten, die Platz verschwenden und Ihre Abfragen verlangsamen. Sie sind Ihrer einen Tabelle treu geblieben, solange es möglich war. Aber die Welt ist groß und weit, und manchmal braucht man **mehrere Tabellen**, um seine Daten festzuhalten, zu steuern und die eigene Datenbank unter Kontrolle zu halten.

Ein Date für Norbert	282
Alles ist verloren ... doch, Moment	293
Über eine Tabelle hinausdenken	294
Die Mehr-Tabellen-Clown-Datenbank	295
Das clown_info-Datenbankschema	296
Aus einer Tabelle zwei machen	298
Ihre Tabellen verknüpfen	303
Ihren Fremdschlüssel beschränken	305
Warum Fremdschlüssel?	306
Eine Tabelle mit einem Fremdschlüssel erstellen	307
Beziehungen zwischen Tabellen	309
Datenmuster: 1:1	309
Datenmuster: Verwendungszwecke für 1:1-Tabellen	310
Datenmuster: eins-zu-viele	311
Datenmuster: nun die m:n-Beziehungen	312
Datenmuster: Wir brauchen eine Verbindungstabelle	315
Datenmuster: m:n	316
Endlich 1NF	321
Schlüssel auf mehreren Spalten	322
Kurzform-Notationen	324
Partielle funktionelle Abhängigkeit	325
Transitive funktionelle Abhängigkeit	326
2. Normalform	330
Die 3. Normalform (endlich)	336
Glücklich bis ans Ende ihrer Tage	339
Ihr SQL-Werkzeugkasten	340




Joins und Mehr-Tabellen-Operationen

8

Vereinigen wir uns!

Willkommen in der Mehr-Tabellen-Welt. Es ist wunderbar, dass Sie jetzt **mehrere Tabellen** in Ihrer Datenbank haben, aber Sie müssen ein paar **neue Werkzeuge und Techniken** kennenlernen, damit Sie mit ihnen auch arbeiten können. Mehrere Tabellen führen schnell zu Verwirrungen, deswegen benötigen Sie **Aliase**, um Ihre Tabellen auseinanderzuhalten. Und **Joins** helfen Ihnen, Ihre Tabellen zu verknüpfen, damit Sie wieder an all die Daten herankommen, die wir im letzten Kapitel verteilt haben. Machen Sie sich fertig, es ist Zeit, die **Steuerung der Datenbank** wieder zu übernehmen.

Immer diese Wiederholungen ...	344
Ihre Tabellen vorab füllen	345
Die »Schwer zu normalisieren«-Misere	347
Die besonderen Interessen (Werte)	348
Interessiert bleiben	349
Alle Interessen aktualisieren	350
Alle Interessen abfragen	351
Viele Wege führen nach Rom	352
CREATE, SELECT und INSERT (fast) gleichzeitig	352
CREATE, SELECT und INSERT gleichzeitig	353
Was das AS macht ...	354
Spaltenaliase	355
Wer braucht schon Tabellenaliase?	356
Alles, was Sie schon immer über Joins wissen wollten ...	357
Der Kartesische Join	358
Ihren Inneren Join befreien	363
Der Inner-Join im Einsatz: der Equi-Join	364
Der Inner-Join im Einsatz: der Nicht-Equi-Join	367
Der letzte Inner: der Natural-Join	368
Vereinte Abfragen?	375
Tabellen- und Spaltenaliase im Gespräch	376
Ihr SQL-Werkzeugkasten	377



... und da kommen die kleinen Ergebnistabellen wirklich her!

Unterabfragen

9

Abfragen in Abfragen

Ja, Hans, mir wäre eine zweiteilige Frage recht. Joins sind wunderbar, aber manchmal müssen Sie *Ihrer Datenbank mehrere Fragen stellen*. Oder **das Ergebnis einer Abfrage nehmen und als Eingabe für eine andere Abfrage verwenden**. An dem Punkt kommen **Unterabfragen** ins Spiel. Sie helfen Ihnen, **doppelte Daten zu vermeiden, und machen Ihre Abfragen dynamischer**. Sie verschaffen Ihnen sogar Zugang zu diesen exklusiven Backstage-Partys. (Na, nicht wirklich, aber zwei von drei ist ja keine so schlechte Quote!)

Gregor wird Jobvermittler	380
Gregors Liste erhält mehr Tabellen	381
Gregor nutzt einen Inner-Join	382
Aber er möchte noch andere Abfragen probieren	384
Unterabfragen	386
Zwei Abfragen zu einer kombinieren	387
Als wäre eine Abfrage nicht genug: die Unterabfrage	388
Eine Unterabfrage im Einsatz	389
Regeln für Unterabfragen	391
Unterabfragenaufbausknekkurs	394
Eine Unterabfrage als SELECT-Spalte	397
Eine Unterabfrage mit einem Natural-Join	398
Eine nicht-korrelierte Unterabfrage	399
SQL im Gespräch: Die beste Form einer Abfrage finden	400
Nicht-korrelierte Unterabfragen mit mehreren Werten	403
Korrelierte Unterabfragen	408
Eine (nützliche) korrelierte Unterabfrage mit NOT EXISTS	409
EXISTS und NOT EXISTS	410
Gregors Jobvermittlung öffnet die Türen	412
Auf dem Weg zur Party ...	413
Ihr SQL-Werkzeugkasten	414

ÄUSSERE Abfrage
INNERE Abfrage

```
SELECT eine_Spalte, andere_Spalte
FROM Tabelle
WHERE Spalte = (SELECT Spalte FROM Tabelle);
```

Äußere Abfrage.

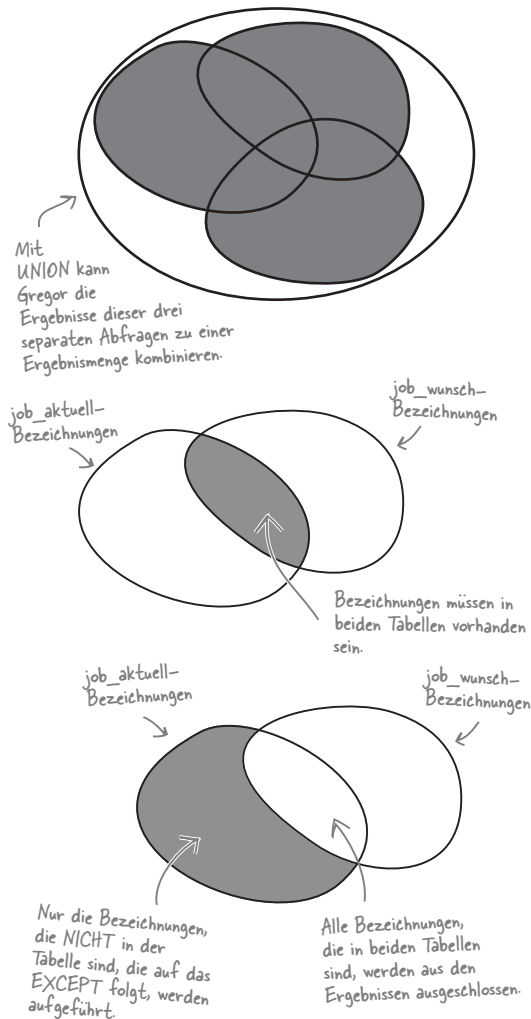
Innere Abfrage.

Outer-Joins, Self-Joins und Unions

Neue Manöver

10

Noch kennen Sie die Join-Geschichte nur halb. Sie wissen, dass Cross-Joins alle Zeilen beider Tabellen liefern und Inner-Joins die Zeilen, die einer Bedingung genügen. Was Sie noch nicht kennen, sind **Outer-Joins**, die Ihnen Zeilen liefern, zu denen es keine passenden Gegenstücke in der anderen Tabelle gibt, **Self-Joins**, die (was ziemlich seltsam erscheint) *eine Tabelle mit sich selbst verknüpfen*, und **Unions**, die die *Ergebnisse Ihrer Abfragen kombinieren*. Haben Sie das alles einmal gelernt, werden Sie Ihre Daten immer so abrufen können, wie Sie sie gerade benötigen. (Und auch die Frage nach der Wahrheit von Unterabfragen haben wir nicht vergessen!)



Alte Daten aufräumen	418
Hier geht's links und rechts	419
Hier ist ein Left-Outer-Join	420
Outer-Joins und mehrere Treffer	425
Der Right-Outer-Join	426
Während Sie mit Outer-Joins gespielt haben ...	429
Wir könnten eine neue Tabelle erstellen	430
Wie sich die neue Tabelle einfügt	431
Ein selbstreferentieller Fremdschlüssel	432
Eine Tabelle mit sich selbst verknüpfen	433
Wir brauchen einen Self-Join	435
Mehr-Tabellen-Informationen, 2. Weg	436
Sie können eine UNION verwenden	437
UNION ist beschränkt	438
UNION-Regeln im Einsatz	439
UNION ALL	440
Aus einer UNION eine Tabelle erstellen	441
INTERSECT und EXCEPT	442
Mit Joins sind wir fertig, Zeit weiterzugehen ...	443
Unterabfragen und Joins im Vergleich	443
Eine Unterabfrage in einen Join umwandeln	444
Ein Self-Join als Unterabfrage	449
Gregors Unternehmen wächst	450
Ihr SQL-Werkzeugkasten	452

Constraints, Views und Transaktionen

Zu viele Köche verderben die Datenbank

11

Ihre Datenbank ist gewachsen und muss jetzt auch von anderen verwendet werden. Das Problem ist, dass sich einige von ihnen mit SQL vielleicht nicht so gut auskennen wie Sie. Sie brauchen Möglichkeiten zu verhindern, **dass sie falsche Daten eingeben**, Techniken, die erlauben, ihnen **nur Teile der Daten zu präsentieren**, und Wege, **sie davon abzuhalten, sich gegenseitig auf die Füße zu treten, wenn sie parallel Daten eingeben**. In diesem Kapitel beginnen wir, unsere Daten vor den Fehlern anderer zu schützen. Willkommen zu Datenbanken in der Defensive, Teil 1.

Gregor hat Hilfskräfte eingestellt	456
Tims erster Tag: Einen neuen Kunden einfügen	457
Tim vermeidet eine NULL	458
Drei Monate später	459
Einen CHECK-Constraint einfügen	460
Das Geschlecht einschränken	461
Franks Job wird lästig	463
Einen View erstellen	465
Views betrachten	466
Was der View tatsächlich macht	467
Was ein View ist	468
Einfügen, Aktualisieren und Löschen mit Views	471
Vorgeben, dass der View eine richtige Tabelle ist	472
Ein View mit CHECK OPTION	475
Ihr View ist aktualisierbar, wenn ...	476
Wenn Sie Ihre Views nicht mehr brauchen	477
Wenn guten Datenbanken Böses widerfährt	478
Was im Geldautomaten abläuft	479
Mehr Ärger am Geldautomaten	480
Es ist kein Traum, es ist eine Transaktion	482
Der klassische ACID-Test	483
SQL hilft Ihnen, Transaktionen zu steuern	484
Was im Geldautomaten hätte passieren sollen	485
Unter MySQL mit Transaktionen arbeiten	486
Und jetzt sind Sie dran	487
Ihr SQL-Werkzeugkasten	490



12

Sicherheit

Ihre Anlagen schützen

Sie haben eine ungeheure Menge an Zeit und Energie in die Erstellung Ihrer Datenbank gesteckt. Und Sie wären erschüttert, wenn dieser etwas passieren würde. Aber Sie mussten auch anderen Personen **Zugriff auf Ihre Daten geben** und sorgen sich jetzt, dass diese etwas falsch einfügen oder aktualisieren könnten oder – noch schlimmer – **die falschen Daten löschen könnten**. Jetzt werden Sie lernen, wie Datenbanken und die Objekte darin **sicherer** gemacht werden und wie Sie die vollständige Steuerung darüber erhalten, **wer was mit Ihren Daten machen darf**.

Anwenderprobleme	494
Fehler in der Clown-Datenbank vermeiden	495
Das Benutzerkonto root schützen	497
Einen neuen Benutzer hinzufügen	498
Entscheiden, was genau ein Benutzer braucht	499
Eine einfache GRANT-Anweisung	500
GRANT-Variante	503
Berechtigungen mit REVOKE widerrufen	504
Eine verwendete GRANT OPTION widerrufen	505
REVOKE mit Präzision	506
Das Problem mit geteilten Konten	510
Die Rolle verwenden	512
Rollen löschen	512
Eine Rolle WITH ADMIN OPTION verwenden	514
CREATE USER und GRANT kombinieren	519
Gregors Liste wird international	520
Ihr SQL-Werkzeugkasten	522
Wie wäre es mit Gregors Liste in Ihrer Stadt?	524
SQL in Ihren eigenen Projekten	524



root



traeger



doc



narr



sauer



froh



muede



krank

Was übrig bleibt



Die Top Ten der Themen, die wir nicht behandelt haben

Aber selbst jetzt sind noch ein paar Dinge übrig. Es gibt noch einiges, das Sie wissen sollten. Wir hätten ein schlechtes Gefühl, wenn wir sie ignorieren würden, auch wenn sie nur eine kurze Erwähnung finden. Bevor Sie das Buch zuklappen, sollten Sie also einen Blick auf diese **kurzen, aber wichtigen SQL-Happen werfen**.

Und wenn Sie das hinter sich haben, bleiben wirklich nur noch zwei weitere Anhänge ... und der Index ... und vielleicht etwas Werbung ... und dann haben Sie es wirklich geschafft. Versprochen!

Nr. 1 Ein GUI für Ihr RDBMS	526
Nr. 2 Reservierte Wörter und Sonderzeichen	528
Nr. 3 ALL, ANY und SOME	530
Nr. 4 Mehr zu Datentypen	532
Nr. 5 Temporäre Tabellen	534
Nr. 6 Datentypen umwandeln	535
Nr. 7 Wer bin ich? Wie spät ist es?	536
Nr. 8 Nützliche mathematische Funktionen	537
Nr. 9 Mit Indizes beschleunigen	539
Nr. 10 PHP/MySQL in 2 Minuten	540

```

Date: Bearbeiten Fenster Hilfe
> SELECT CURRENT_DATE;
+-----+
| CURRENT_DATE |
+-----+
| 2007-07-26   |
+-----+
1 row in set (0.00 sec)
    
```

```

Date: Bearbeiten Fenster Hilfe
> SELECT CURRENT_TIME;
+-----+
| CURRENT_TIME |
+-----+
| 11:26:48     |
+-----+
1 row in set (0.00 sec)
    
```

```

Date: Bearbeiten Fenster Hilfe
SELECT CURRENT_USER;
+-----+
| CURRENT_USER |
+-----+
| root@localhost |
+-----+
1 row in set (0.00 sec)
    
```

>	Größer als	!>	Nicht größer als	>=	Größer gleich
<	Kleiner als	!<	Nicht kleiner als	<=	Kleiner gleich
=	Gleich	<>	Ungleich	!=	Nicht gleich

MySQL-Installation

Probieren Sie es selbst

B

Ihre gesamten neuen SQL-Fertigkeiten bringen Ihnen nicht viel, wenn Sie sie nirgendwo anwenden können.

Dieser Anhang enthält Anleitungen für die Installation Ihres MySQL-RDBMS, mit dem Sie arbeiten können.



Anfangen, und zwar schnell	544
Anleitung und Problembhebung	544
Schritte zur Installation von MySQL unter Windows	545
Schritte zur Installation von MySQL unter Mac OS X	548

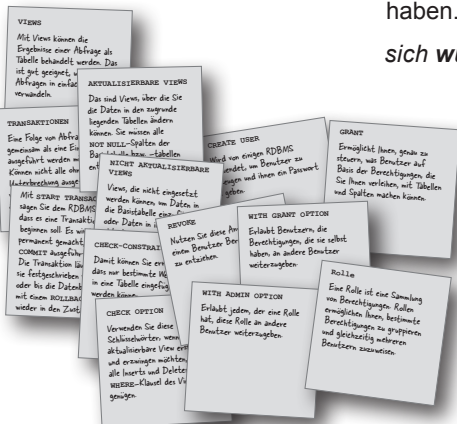
Werkzeug-Überblick

All Ihre SQL-Werkzeuge

C

Hier sind all Ihre SQL-Werkzeuge – das erste Mal an einem Ort und nur für eine Nacht (ist bloß ein Scherz)!

Das ist eine Zusammenfassung all der SQL-Werkzeuge, die wir behandelt haben. Nehmen Sie sich einen Moment, *um einen Blick darauf zu werfen und sich wunderbar zu fühlen – all das haben Sie gelernt!*



Symbole	552
A	552
B, C	553
D-F	554
G-M	555
N-S	556
T-U	557
V-Z	558