

# Database Manager

## User Guide



© 2005 Linux Web Host. All rights reserved. The content of this manual is furnished under license and may be used or copied only in accordance with this license. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without prior written permission of Linux Web Host.

Doc ID: DBMANGR.003 (3/31/05)

# Contents

Getting Started	<b>Chapter 1</b>	
	Introduction .....	1
	About this guide .....	1
	Looking at the interface .....	1
	Creating databases .....	2
Building Tables	<b>Chapter 2</b>	
	Overview .....	3
	Creating table structure .....	3
	Inserting rows .....	4
	Setting column indexes .....	4
	Altering table structure .....	5
	Deleting records .....	5
	Emptying tables .....	6
	Dropping tables .....	6
Browsing Records	<b>Chapter 3</b>	
	Overview .....	8
	Toggling full text display .....	8
	Applying table layouts .....	9
	Setting result limit preference .....	9
	Sorting table rows .....	10
	Searching .....	10
Performing Table Operations	<b>Chapter 4</b>	
	Overview .....	11
	Moving tables .....	11
	Copying tables .....	11
	Performing table maintenance .....	12
	Setting table type .....	12
Managing Users	<b>Chapter 5</b>	
	Introduction .....	14
	About user roles .....	14
	Adding users .....	14
	Setting up an ODBC connection .....	15

Exporting

## Chapter 6

Overview .....	16
About export formats .....	16
Exporting databases .....	16
Exporting tables .....	19

# Chapter 1: Getting Started

## Introduction

Welcome to DB Manager, a powerful tool that allows you to create, manage, and administer MySQL databases on your website. Built upon the popular open source utility, *phpMyAdmin*, DB Manager includes an intuitive GUI that lets both novice users and seasoned database administrators work quickly and efficiently.

## About this guide

This guide is a comprehensive resource covering all features of DB Manager. After reading it, you will know how to:

- create databases (either from scratch or by importing data from another source)
- delete databases
- design and populate tables
- set up user accounts so that others can access your data
- export databases

You will also gain a basic understanding of *SQL*, or *Standard Query Language*. SQL is a command language consisting of keywords like SELECT, INSERT and DROP and the syntax governing how they may be used (SQL statements are also known as *queries*). You can type queries directly or use the GUI to build them visually; regardless of the method you choose, DB Manager displays the queries generated as feedback.

## Looking at the interface

The Database Manager GUI consists of two main parts: the *main window*, which you use to create databases, users, and ODBC connections; and the *phpMyAdmin* window, the utility you use to manage a selected database.



Figure 1: The Main interface (left) and phpMyAdmin window (right) of the Database Manager GUI.

### To access Database Manager:

Click the Database Manager icon on your Control Panel.

## Creating databases

Database Manager is installed with a default database that uses the same name as your domain name (with underbars replacing periods). For example, if your domain is *mydomain.com*, your default database will be *mydomain\_com*. As you create more databases, the default name (with a trailing underbar) is used as a prefix for their names, for example, *mydomain\_com\_contacts*. Once you have created a database, you can build the tables associated with it in phpMyAdmin.

### To create a database:

- 1 Open Database Manager by clicking its icon in the control panel.
- 2 Click Create New Database. An input field appears.

**Create New Database**

Database Name	<input type="text" value="mydomain_com_"/>
<input type="button" value="Add"/> <input type="button" value="Clear"/> <input type="button" value="Back"/>	

- 3 Enter the suffix part of the name.
- 4 Click Add.

**Note:** To clear what you have entered in the field, click Clear instead of Add; to abandon creation and return to DBManager Home, click the Back button.

### To delete a database:

- 1 Click the radio button to the left of the database you want to delete.
- 2 Click Delete. Confirm the deletion.

# Chapter 2: Building Tables

## Overview

Tables are the basic building blocks of databases. They consist of *columns* (or *fields*) and *rows*. Each row is a single data record; for example, if you had a table for storing customer names, each row might contain a customerID field, a first\_name field and a last\_name field. You use phpMyAdmin to create, alter and delete records from tables.

## Creating table structure

Before you can insert data into table, you must create its structure. The structure of a table is determined by its columns, with each column in the table set up to handle a particular kind of data. Each column has specific attribute settings

column	description
field	the name of the column
type	the data type of the column
length/values	the appropriate length/values setting for the selected data type
attributes	any optional attribute you wish to apply (for example, UNSIGNED to only allow zero and positive numbers.)
null	a keyword which determines if the field for the record can be left blank, set to either NULL (can be empty) or NOT NULL (cannot be empty)
default	a default value for the column (if one is not provided when the record is inserted into the table)
extra	use AUTO_INCREMENT to automatically increment the value in an ID field.

### To create a table:

- 1 In DBManager Home, select the database into which you want to add the table.
- 2 Click the Manage Database button.
- 3 Enter a name for the table, and specify the number of fields (columns) it will contain. Click Go.

4 Enter column information.

5 Click Save.

phpMyAdmin refreshes and shows the table structure as well as the SQL query used to create it; you are now ready to insert rows (records) into your table.

## Inserting rows

The rows of a table store its actual information (data). Adding rows into a table is similar to adding rows into a spreadsheet.

### To insert a row:

1 Select the table into which you want to insert a row. If you have just created the table structure, the table will be preselected; to work with a different table, select it from the list of tables (in the left pane of the phpMyAdmin interface).

2 Click the Insert tab.

3 Enter values for the each column in the row. If a column is set to `auto_increment` (for example, if you are using a numbered ID field) leave the column blank.

4 Click Go to add the row or Reset to clear the values you entered and try again.

**Note:** To continue inserting rows without returning to table structure view, select the Insert another row radio button before clicking Go.

## Setting column indexes

Databases typically include several tables and use special flags—called *indexes* or *keys*—to reference these tables, ensure uniqueness of data or optimize text search queries. The four types of column indexes (along with their associated phpMyAdmin Action buttons) are:



**Primary** The identifying column of a table. Each table may only contain one primary key.



**Index** Applied to a key to improve the performance of queries involving it. You can set as many indexes as you wish.



**Unique** A key set on a column to ensure that no two rows in the table use the same value for the field.



**Full Text** A key you set on a text field to allow it to be searched.

### To set a column index:

- 1 In phpMyAdmin, select the table which contains the column on which you want to set an index.
- 2 Click the Action button for the type of column index you want to set (primary, index, unique or full text). The feedback box refreshes to describe your action and show you the SQL query that made it happen.

## Altering table structure

You can alter the structure of a table you have created. You can add, delete (drop), or rename a column, or change its attributes.

### To alter a column:

- 1 Select the table you want to work with from the table list pane.



- 2 Click the Change button (✎) for the column you want to alter. To select more than one column, check the checkboxes next to the columns you want to work with and then click the Change button to the right of the *With Selected* label. To select all columns, click the Check All link and then click the Change button to the right of the *With Selected* label.
- 3 Edit the attributes as desired.
- 4 Click Save.

### To delete a column:

- 1 Click the Drop button (🗑) for the column you want to delete. To delete more than one column, check the checkboxes next to the columns you want to delete and then click the Delete button to the right of the *With Selected* label.
- 2 At the prompt, click OK to confirm deletion.

**Note:** You cannot delete all columns from a table (because you cannot have a table without columns.)

## Deleting records

To remove actual data from a table (for example, to remove selected customers from a customer table) you delete the relevant records.

### To delete a record:

- 1 Select the table you want to work with from the table list pane.
- 2 Click the Browse button.

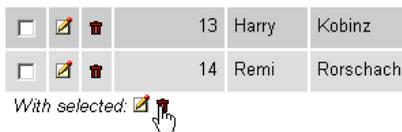


- 3 Check the checkboxes of the records you want to delete.



	customerID	first	last
<input checked="" type="checkbox"/>	2	Wilson	Wonderful
<input checked="" type="checkbox"/>	3	Clara	Ruckstuhl
<input type="checkbox"/>	4	William	Burroughs
<input type="checkbox"/>	5	Frank	Truffaut
<input type="checkbox"/>	6	Gaspard	Winckler

- 4 Click the Drop button at the bottom of the table.



## Emptying tables

When you need to remove all records from a table, you *empty* it. Emptying a table removes the data but preserves the structure.

### To empty a table:

- 1 Select the table you want to work with from the table list pane.



- 2 Confirm that you want to empty the table.

## Dropping tables

In the namespace of SQL, you do not *delete* tables or databases, you *drop* them. When you drop a table, you permanently delete both its structure and its data.

**To drop a table:**

- 1 Select the database you want to work with from DBManager Home.
- 2 Click Manage Database. phpMyAdmin launches.
- 3 Click the database name above the table list.



- 4 Click the Drop button for the table you want to get rid of.

Table	Action	Records	Type	Size	Overhead
<input type="checkbox"/> customers	     	13	MyISAM	4.3 KB	-
<input type="checkbox"/> orders	     	0	MyISAM	1.0 KB	-
<input type="checkbox"/> products	     	5	MyISAM	2.1 KB	-
3 table(s)	Sum	18	--	7.4 KB	0 Bytes

Check All / Uncheck All      With selected: ▾

# Chapter 3: Browsing Records

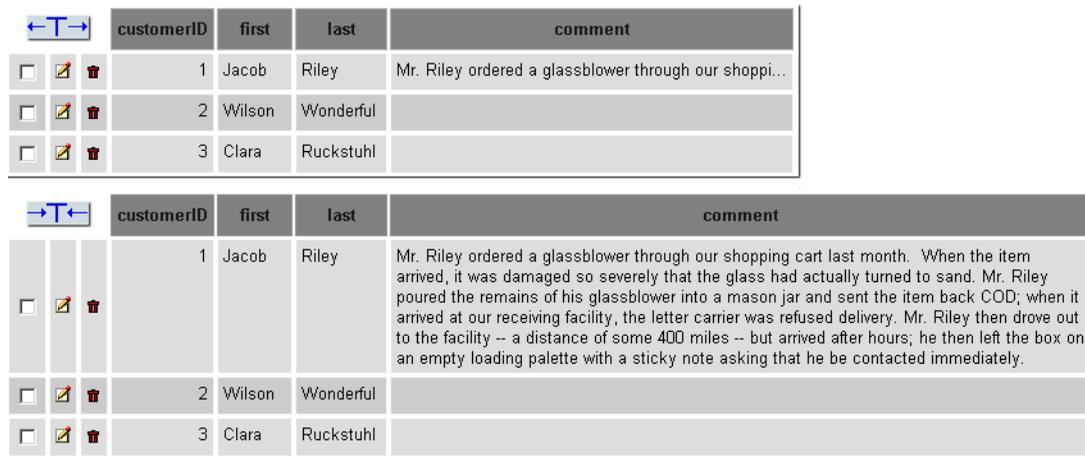
## Overview

You can control how tables are displayed when you browse them in phpMyAdmin. Specifically, you can:

- toggle full text or partial text views of records containing long text entries.
- apply a different table layout.
- set a result limit (to control how many records appear per screen).
- change how rows are sorted.

## Toggle full text display

If the table you are browsing includes a text type field (e.g., LONGTEXT), you can choose to expand the field (to show full text) or collapse the field (to show partial text and conserve screen real estate).



		customerID	first	last	comment
<input type="checkbox"/>		1	Jacob	Riley	Mr. Riley ordered a glassblower through our shoppi...
<input type="checkbox"/>		2	Wilson	Wonderful	
<input type="checkbox"/>		3	Clara	Ruckstuhl	

		customerID	first	last	comment
<input type="checkbox"/>		1	Jacob	Riley	Mr. Riley ordered a glassblower through our shopping cart last month. When the item arrived, it was damaged so severely that the glass had actually turned to sand. Mr. Riley poured the remains of his glassblower into a mason jar and sent the item back COD; when it arrived at our receiving facility, the letter carrier was refused delivery. Mr. Riley then drove out to the facility -- a distance of some 400 miles -- but arrived after hours; he then left the box on an empty loading palette with a sticky note asking that he be contacted immediately.
<input type="checkbox"/>		2	Wilson	Wonderful	
<input type="checkbox"/>		3	Clara	Ruckstuhl	

Figure 2: A LONGTEXT type field named comment, shown with partial text (top) and full text (below).

### To toggle full text display:

- 1 Select the table you want to work with in phpMyAdmin. (Select DBHome > database\_name > Manage Database, then make a selection from the table list.)
- 2 Click the Browse tab.

Do one of the following:

- To show full text click the Full Text button.
- To show partial text click the Partial Text button.

## Applying table layouts

Depending upon the type of data you are storing in a table, you may want to change the visual layout of the table to make it easier to read. You can choose from three layout modes:

- horizontal (default)
- horizontal with rotated headers
- vertical

**HORIZONTAL**

	customerID	first	last
<input type="checkbox"/>	6	Gaspard	Winckler
<input type="checkbox"/>	7	Joseph	Nieto
<input type="checkbox"/>	8	Robert	Peel
<input type="checkbox"/>	9	Harrow	Greenhill
<input type="checkbox"/>	10	Keith	Neverdie

**VERTICAL**

customerID	1	2	3	4	5
first	Jacob	Wilson	Clara	William	Frank
last	Riley	Wonderful	Ruckstuhl	Burroughs	Truffaut

**HORIZONTAL (ROTATED HEADERS)**

	customerID	first	last
<input type="checkbox"/>	11	Arnold	Layne
<input type="checkbox"/>	12	Paco	Domingo
<input type="checkbox"/>	13	Harry	Kobinz
<input type="checkbox"/>	14	Remi	Rorschach

Figure 3: Table browsing layouts (horizontal, vertical and horizontal with rotated headers).

### To apply a table layout:

- 1 Select the desired layout mode from the mode option list.
- 2 Click the Show button.

### Setting result limit preference

A *result limit* defines which row to begin with and how many rows to show per page when you are browsing a table in phpMyAdmin. (The word *limit* refers to the LIMIT keyword in the query display box displayed above the table).

### To set result limit preference:

- 1 Enter the number of rows to show and the starting row number.
- 2 Click Show.

## Sorting table rows

You can sort a table by column (in ascending or descending order) for any column that is indexed; an auto\_increment column (such as a customerID field) will always be indexed.

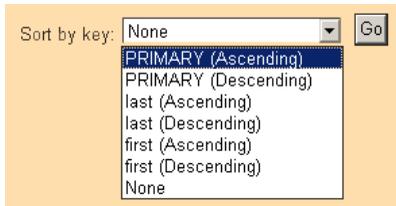


Figure 4: Sort options for a customers table

### To set sorting option:

- 1 Select a sort option from the *Sort by key* dropdown list.
- 2 Click Go.

If a field you want to sort on does not appear in the dropdown list, this means it has not yet been indexed. For information on indexing columns, see [Setting column indexes](#).

## Searching

You can use phpMyAdmin to search for a particular record or records in a table. You can set specific search parameters for each column in the table, set which field the results will be sorted on and how they will be displayed. For integer fields (columns that store numeric values) you can search for equal (=), not equal (!=), greater (>), lesser (<), greater-or-equal (=>) or lesser-or-equal (=<). For text fields (for example, the VARCHAR data type) you can search for text that matches (=) or does not match. Both integer and text fields also support the LIKE keyword; to use this option, enclose your search string between percentage signs (%), for example, %456% or %Rob%.

### To search a table:

- 1 Select the table you want to work with from the table list pane.
- 2 Click the Search button.



- 3 Set search options for each field.
- 4 Click Go.

# Chapter 4: Performing Table Operations

## Overview

A *table operation* is an action you perform on a table—moving or copying it to another database, reordering its columns, or renaming it. You can also use table operations to perform maintenance on a table, for example, restoring a table that you suspect has been corrupted.

## Moving tables

When you move a table, you remove it from one database and add it to another as-is. You can not overwrite an existing table in another database with the table you are moving.

### To move a table:

- 1 Select the table you want to work with from the table list pane.



- 2 Click the Operations button.



- 3 Enter a name for the table in the Move table to input field. By default, the current name is used.

- 4 Click Go.

## Copying tables

When you copy a table, you have a number of considerations to make. You can choose to:

- copy just the structure (columns), just the data (records) or both (structure and data).
- include a “drop table” statement in the SQL query phpMyAdmin sends to the database; this will overwrite an existing table that has the same name as the copied table.
- check the `auto_increment` option to append the records of the copied table to the existing table; the starting `auto_increment` value (typically an ID field) will be one increment above the highest value in the existing table.
- switch to the copied table once the action has been performed.

### To copy a table:

- 1 Enter a name for the table in the Copy table to input field. By default, the current name is used.
- 2 Check the desired options.
- 3 Click Go.

## Performing table maintenance

Table maintenance operations are advanced functions. You should not use them without first reading and understanding the MySQL documentation.

**Checking** is used to determine if a table contains errors; it runs a 'myisamchk -m' on 'MyISAM' and 'InnoDB' files from within MySQL. For 'MyISAM' tables the key statistics are updated. You can get many rows of information for each checked table. The last row will be of 'Msg\_type status' and should normally be 'OK'. If you don't get 'OK', or 'Not checked' you should run a repair of the table.

**Analyzing** is used to analyze and store the key distribution for the table. During the analysis, the table is locked with a read lock. This works on 'MyISAM' and 'BDB' tables and is equivalent to running 'myisamchk -a' on the table. MySQL uses the stored key distribution to decide in which order tables should be joined when one does a join on something other than a constant.

**Repairing** is used to fix a table that you suspect may have been corrupted. Executing this operation returns a message describing whether a repair was warranted, and what exactly the problem was. Do not attempt to repair a table without first checking it.

**Optimizing** is used to reclaim unused space and defragment the data file. 'OPTIMIZE TABLE' works only on 'MyISAM' and 'BDB' tables. For 'BDB' tables, 'OPTIMIZE TABLE' is mapped to 'ANALYZE TABLE'.

**Flushing** is used to clear some of the internal caches MySQL uses. The flush commands include actions against: HOSTS, DES\_KEY\_FILE, LOGS, PRIVILEGES, QUERY CACHE, and TABLES. You can run these commands through the mysqladmin utility using 'flush-hosts', 'flush-logs', 'reload', or 'flush-tables' commands.

## Setting table type

You can choose to set the table type of the current table; by default, the format will be MyISAM. Setting table type is another advanced option.

### To set table type:

- 1 Select the desired table type from the dropdown list.

**2** Click Go.

# Chapter 5: Managing Users

## Introduction

Database Manager supports a sophisticated *privilege system* that grants and restricts privileges to database users based on their *roles*. You create user accounts (and associated *ODBC connections*) to allow users to connect remotely to your databases and perform actions on them, typically through a web page which includes a connection script written in PHP.

## About user roles

Each user role allows certain MySQL commands to be executed. There are four user roles, three of which you may assign to the users you create:

**Database Administrator** The default user created with each new database; you as the DSM owner are the Database Administrator. This user role allows full administrative access to the database (MySQL commands: SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, GRANT, INDEX, ALTER, CREATE TEMPORARY TABLE, and LOCK TABLES). The Database Administrator User cannot be deleted or edited.

**Read** Allows a user to browse a database (MySQL command: SELECT).

**Read/Write** Allows a user to browse a database, insert new records, update records and delete records (MySQL commands: SELECT, UPDATE, ALTER and DROP).

**Full Access** Gives user the same privileges as the Database Administrator except GRANT.

## Adding users

User accounts can be mapped to specific people (e.g., an employee responsible for updating a product table) or to the general public (e.g. visitors to your website who browse the product table). Each database has its own set of users.

### To add a user:

- 1 From DBManager Home, click the database for which you want to add a user. The User list and ODBC list are displayed.
- 2 Click the Add User button.
- 3 Enter a username and password. Confirm the password.
- 4 Select the Role Type for the user.
- 5 Click OK.

**To change a user's role:**

- 1 Check the radio button to the left of the user you want to delete.
- 2 Click the Modify User button.
- 3 Select a new role for the user.
- 4 Click Update.

**To delete a user:**

- 1 Check the radio button to the left of the user you want to delete.
- 2 Click the Delete User button. Confirm the deletion.

**Note:** An ODBC connection associated with the deleted user will be deleted, too.

### **Setting up an ODBC connection**

ODBC (Open DataBase Connectivity) is an open standard which allows virtually any system or application to access a database. You can set up an ODBC for users to allow them to connect to your database from a remote location.

**To setup an ODBC connection:**

- 1 From DBManager Home, click the database to which you want to establish an ODBC connection.
- 2 Click Create ODBC.
- 3 Select a user for the ODBC connection from the dropdown list.
- 4 Enter an IP Address. If you want to allow an entire C class to use an ODBC connection, check the checkbox indicating so.
- 5 Click Save.

# Chapter 6: Exporting

## Overview

Database Manager allows you to export a database (or selected tables) to a file. Some file formats allow you to view the records in another application, such as Microsoft Excel. The SQL format is used for backup/restoration purposes; it creates an SQL file (a series of queries) that can be imported back into MySQL to restore a database.

## About export formats

Database Manager can export databases (or tables) in the following formats:

**LaTeX** A document preparation format for high-quality typesetting.

**CSV** Comma Separated Values. A text file in which columns are delimited by commas (or semicolons) and records by line breaks.

**CSV for MS Excel** A CSV file optimized for use in Microsoft Excel.

**XML** Extensible Markup Language. A flexible text format derived from SGML (Standard Generalized Markup Language).

**SQL** Standard Query Language. The command language used to administer databases.

## Exporting databases

When you export a database, you set options related to the export format you chose. Once you have set these options, you can select the tables you want to include and export the file.

## SQL export options

When you export SQL, you can choose to export the structure (columns), data (records) or both. The structure options are:

- inclusion of a “drop table” statement; if this option is checked, restoring the database will drop (delete) any existing database with the same name.
- addition of an “auto\_increment” value.
- enclosing tables and field names with back quotes.
- adding creation, update and check information to the comments section.

And the data options are:

- setting export type (INSERT, UPDATE or REPLACE)
- use complete inserts.

- use extended inserts.
- use delayed inserts.

**To export an SQL file:**

- 1 Select the database you want to work with from DBManager Home.
- 2 Click Manage Database. phpMyAdmin launches.
- 3 Click the database name above the table list.



- 4 Click the Export button.

**Database *mydomain\_com* running on localhost**



- 5 Select the tables you want to include.
- 6 Set the SQL Structure and Data options as desired.
- 7 Check the Save as file checkbox.
- 8 Click Go.

**LaTeX export options**

LaTeX, like SQL, allows you to set structure and data export options. The structure options control table captioning and key labelling; the data options include column labelling particulars and replace NULL string.

**To export a LaTeX file:**

- 1 Select the desired tables.
- 2 Select LaTeX from the Export box.
- 3 Set the desired LaTeX Options.
- 4 Check the Save as file checkbox.
- 5 Click Go.

**CSV export options**

The default values for CSV export are:

<b>option</b>	<b>default</b>
Fields terminated by	semicolon (;)
Fields enclosed by	quotation marks (“
Fields escaped by	backslashes (\)
Lines terminated by	\r\n
Replace NULL by	NULL
Put field names in row	(checked)

**To export a CSV file:**

- 1 Select the desired tables.
- 2 Select CSV from the Export box.
- 3 Set the desired CSV Options.
- 4 Check the Save as file checkbox.
- 5 Click Go.

**CSV for Microsoft Excel export options**

There are only three CSV for Microsoft Excel export options: replacement string for NULL values, put field names at first row, and specify Excel platform (Windows or Macintosh).

**To export a CSV for Microsoft Excel file:**

- 1 Select the desired tables.
- 2 Select CSV for MS Excel from the Export box.
- 3 Set the desired export options.
- 4 Check the Save as file checkbox.
- 5 Click Go.

**XML export options**

Other than the ability to select which tables to export, there are no XML options.

**To export an XML file:**

- 1 Select the desired tables.

- 2 Select XML from the Export box.
- 3 Check the Save as file checkbox.
- 4 Click Go.

## Exporting tables

The export options for tables are the same as the database options, with one difference: when you are exporting a single table, you can specify the range of records to export.

### To export a table:

- 1 Select the table you want to work with from the table list pane.



- 2 Click the Export tab.



- 3 Set the desired Export options.
- 4 Specify the number of records to dump (write to the export file) and the starting record number.
- 5 Click Go.