Chapter No. 10
"Table and Database Operations"
In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.10 "Table and Database Operations"

A synopsis of the book’s content

Information on where to buy this book

About the Author

Marc Delisle is a member of the MySQL Developers Guild— which regroups community developers— because of his involvement with phpMyAdmin. He started to contribute to this popular MySQL web interface in December 1998, when he made the first multi-language version. He has been actively involved with this software project since May 2001 as a developer and project administrator.

Marc has worked since 1980 at Collège de Sherbrooke, Québec, Canada, as an application programmer and network manager. He has also been teaching networking, security, and PHP/MySQL application development. In one of his classes, he was pleased to meet a phpMyAdmin user from Argentina.

Marc lives in Sherbrooke with his wife and they enjoy spending time with their four children.

This book is Marc's first one, and was followed by "Creating your MySQL Database: Practical Design Tips and Techniques", also with Packt Publishing.

I am truly grateful to Louay Fatoohi who approached me for this book project, and to the Packt team whose sound comments were greatly appreciated during the production. My thanks also go to Garvin Hicking, Alexander Marcus Turek, and Kai 'Oswald' Seidler, the reviewers for the successive editions of this book. Their sharp eyes helped in making this book clearer and more complete.

Finally, I wish to thank all contributors to phpMyAdmin's source code, translations, and documentation; the time they gave to this project still inspires me and continues to push me forward.

Mastering phpMyAdmin 2.11 for Effective MySQL Management

Providing a powerful graphical interface for managing MySQL, phpMyAdmin is one of the most popular open source applications. While most MySQL developers use routine features of phpMyAdmin every day, few are aware of the power and potential of its advanced features. This book builds a solid understanding of the core capabilities of phpMyAdmin before walking you through every facet of this legendary tool.

Used by millions of developers, MySQL is the most popular open source database, supporting numerous large dynamic websites and applications. MySQL has acquired this wide popularity by virtue of its open source nature, performance, reliability, robustness, and support for various platforms. However, this popularity has also been helped by the existence of phpMyAdmin, the industry standard administration tool that makes database management easy for both the experienced developer and their novice.

The powerful graphical interface that it provides to MySQL has made phpMyAdmin an indispensable tool for MySQL and web developers. Every phpMyAdmin user can benefit from unlocking the full potential of this powerful application.

What This Book Covers

Chapter 1 will take a look at how the Web has evolved as a means to deliver applications and why we should use PHP/MySQL to develop these applications. We'll also take a look at how phpMyAdmin is recognized as a leading application to interface MySQL from the Web, the history of phpMyAdmin, and a brief list of its features.

Chapter 2 will take a look at the common reasons for installing phpMyAdmin, the steps for downloading it from the main site, the basic configuration, and uploading it to our web server. We will also learn how to use a single copy of phpMyAdmin to manage multiple servers, and the usage of authentication types to fulfill the needs of a users' group while protecting authentication credentials.

Chapter 3 will take a look at the phpMyAdmin interface in detail.

Chapter 4 will cover how to create a database and tables, and how to enter data, manually, in the tables. We'll also see how to confirm the presence of data by using the browse mode; including the SQL query links, navigation bar, sorting options, and row marking.
Chapter 5 will examine concepts like editing data, including the null field and using the Tab key, applying a function to a value, duplicating rows of data, and deleting data, tables, and databases.

Chapter 6 will cover how to add fields, including special field types like TEXT, BLOB, ENUM, and SET, how to use a calendar popup for DATE, DATETIME, and TIMESTAMP fields, and how to upload binary data into a BLOB field. We'll also see how to manage indexes (multi-field and full-text) and get feedback from MySQL about which indexes are used in a specific query.

Chapter 7 will examine the various ways to trigger an export: from the Database view, the Table view, or a results page. We'll also take a look at the various available export formats, their options, the possibility of compressing the export file, and the various places where it might be sent.

Chapter 8 will look at the various options in phpMyAdmin that allow us to import data, the different mechanisms involved in importing SQL and CSV files, the limits that we might hit when trying a transfer, and ways to bypass these limits.

Chapter 9 will cover single-table searches with query by example criteria and additional criteria specification, selecting displayed values, and ordering results. We'll also look at wildcard searches and full database search.

Chapter 10 will cover the operations we can perform on whole tables or databases. We'll take a look at table maintenance operations for table repair and optimization, changing various table attributes, table movements, including renaming and moving to another database, and multi-table operations.

Chapter 11 will take a look at the installation of the necessary infrastructure for keeping special metadata (data about tables), and we'll learn how to define relations between both InnoDB and non-InnoDB tables. We'll also examine the modified behavior of phpMyAdmin when relations are present, foreign keys, getting information from the table, the Designer feature, and column-commenting.

Chapter 12 will take a look at the purpose of query boxes and where they can be found. We'll also look at query window options, multi-statement queries, how to use the field selector, how to use the SQL Validator, and how to get a history of the typed commands.
Chapter 13 will cover various aspects such as opening the query generator, choosing tables, entering column criteria, sorting and showing columns, and altering the number of criteria rows or columns. We'll also see how to use the AND and OR operators to define relations between rows and columns, and how to use automatic joins between tables.

Chapter 14 will take a look at how to record bookmarks (after or before sending a query), how to manipulate them, and how some bookmarks can be made public. We'll learn about the default initial query for Browse mode. We'll also see passing parameters to bookmarks and executing bookmarks directly from the pma_bookmark table.

Chapter 15 will cover the documentation features offered by phpMyAdmin—the print view for a database or a table, and the data dictionary for a complete column list. We'll also see PDF relational schemas.

Chapter 16 will examine how we can improve the browsing experience by transforming data using various methods. We'll see thumbnail and full-size images of .jpeg and .png BLOB fields, generate links, format dates, display only parts of texts, and execute external programs to reformat each cell's contents.

Chapter 17 will cover the use of language files in phpMyAdmin. We'll look at UTF-8 and the impact of switching from one character set to another. We also see how phpMyAdmin has to recode data when the version of MySQL is earlier than 4.1.x, and take a look at the character set and collation features of MySQL version 4.1.x and later.

Chapter 18 will cover how phpMyAdmin permits to manage the main features of MySQL 5.0.

Chapter 19 will discuss how a system administrator can use the phpMyAdmin server management features for day-to-day user account maintenance, server verification, and server protection.

Chapter 20 proposes guidelines for solving some common problems, and gives hints on how to avoid them. It also explains how to interact with the development team for support, bug reports, and contributions.
In the previous chapters, we dealt mostly with table fields. In this chapter, we will learn how to perform some operations that influence tables or databases as a whole. We will cover table attributes and how to modify them, and also discuss multi-table operations.

Various links that enable table operations have been put together on one sub-page of the Table view: Operations. Here is an overview of this sub-page:

**Table and Database Operations**

Table and Database Operations

Table Maintenance
During the lifetime of a table, it repeatedly gets modified, and so grows and shrinks. Outages may occur on the server, leaving some tables in a damaged state.

Using the Operations sub-page, we can perform various operations, but not every operation is available for every table type:

- **Check table**: Scans all rows to verify that deleted links are correct. Also, a checksum is calculated to verify the integrity of the keys; we should get an 'OK' message if everything is all right.

- **Analyze table**: Analyzes and stores the key distribution; this will be used on subsequent `JOIN` operations to determine the order in which the tables should be joined.

- **Repair table**: Repairs any corrupted data for tables in the **MyISAM** and **ARCHIVE** engines. Note that the table might be so corrupted that we cannot even go into Table view for it! In such a case, refer to the *Multi-Table Operations* section for the procedure to repair it.

- **Optimize table**: This is useful when the table contains overheads. After massive deletions of rows or length changes for **VARCHAR** fields, lost bytes remain in the table. phpMyAdmin warns us in various places (for example, in the Structure view) if it feels the table should be optimized. This operation is a kind of defragmentation for the table. In MySQL 4.x, this operation works only on tables in the **MyISAM**, **Berkeley (BDB)**, and **InnoDB** engines. In MySQL 5.x, it works only on tables in the **MyISAM**, **InnoDB**, and **ARCHIVE** engines.

- **Flush table**: This must be done when there have been lots of connection errors and the MySQL server blocks further connections. Flushing will clear some internal caches and allow normal operations to resume.

- **Defragment table**: Random insertions or deletions in an InnoDB table fragment its index. The table should be periodically defragmented for faster data retrieval.

The operations are based on the underlying MySQL queries available—phpMyAdmin is only calling those queries.

Changing Table Attributes
Table attributes are the various properties of a table. This section discusses the settings for some of them.
Table Type

The first attribute we can change is called **Table storage engine**:

This controls the whole behavior of the table: its location (on-disk or in-memory), the index structure, and whether it supports transactions and foreign keys. The drop-down list may vary depending on the table types supported by our MySQL server.

Changing the table type may be a long operation if the number of rows is large.

Table Comments

This allows us to enter comments for the table.
Table and Database Operations

These comments will be shown at appropriate places (for example, in the left panel, next to the table name in the Table view and in the export file). Here is what the left panel looks like when the $cfg['ShowTooltip'] parameter is set to its default value of TRUE:

![Table and Database Operations](image)

The default value of $cfg['ShowTooltipAliasDB'] and $cfg['ShowTooltipAliasTB'] (FALSE) produces the behavior we have seen earlier: the true database and table names are displayed in the left panel and in the Database view for the Structure sub-page. Comments appear when the mouse pointer is moved over a table name. If one of these parameters is set to TRUE, the corresponding item (database names for DB and table names for TB) will be shown as the tooltip instead of the names. This time, the mouse-over shows the true name for the item. This is convenient when the real table names are not meaningful.

There is another possibility for $cfg['ShowTooltipAliasTB']: the 'nested' value. Here is what happens if we use this feature:

- The true table name is displayed in the left panel.
- The table comment (for example project__) is interpreted as the project name and is displayed as such. (See the Nested Display of Tables Within a Database section in Chapter 3).

### Table Order

When we Browse a table or execute a statement such as SELECT * from book, without specifying a sort order, MySQL uses the order in which the rows are physically stored. This table order can be changed with the Alter table order by dialog. We can choose any field, and the table will be reordered once on this field. We choose author_id in the example, and after we click Go, the table gets sorted on this field.

Reordering is convenient if we know that we will be retrieving rows in this order most of the time. Moreover, if later we use an ORDER BY clause and the table is already physically sorted on this field, the performance should be higher.
This default ordering will last as long as there are no changes in the table (no insertions, deletions, or updates). This is why phpMyAdmin shows the (singly) warning.

After the sort has been done on author_id, books for author 1 will be displayed first, followed by the books for author 2, and so on. (We are talking about a default browsing of the table without explicit sorting.) We can also specify the sort order: Ascending or Descending.

If we insert another row, describing a new book from author 1, and then click Browse, the book will not be displayed along with the other books for this author because the sort was done before the insertion.

### Table Options

Other attributes that influence the table's behavior may be specified using the Table options dialog:

The options are:

- **pack_keys**: Setting this attribute results in a smaller index; this can be read faster but takes more time to update. Available for the MyISAM storage engine.
- **checksum**: This makes MySQL compute a checksum for each row. This results in slower updates, but easier finding of corrupted tables. Available for MyISAM only.
Table and Database Operations

- **delay_key_write**: This instructs MySQL not to write the index updates immediately but to queue them for later, which improves performance. Available for MyISAM only.
- **auto-increment**: This changes the auto-increment value. It is shown only if the table’s primary key has the auto-increment attribute.

Renaming, Moving, and Copying Tables

The **Rename** operation is the easiest to understand: the table simply changes its name and stays in the same database.

The **Move** operation (shown in the following screen) can manipulate a table in two ways: change its name and *also* the database in which it is stored:

Moving a table is not directly supported by MySQL, so phpMyAdmin has to create the table in the target database, copy the data, and then finally drop the source table.

The **Copy** operation leaves the original table intact and copies its structure or data (or both) to another table, possibly in another database. Here, the **book-copy** table will be an exact copy of the **book** source table. After the copy, we will stay in the Table view for the **book** table unless we selected **Switch to copied table**.

The **Structure only** copy is done to create a test table with the same structure.
Appendng Data to a Table

The Copy dialog may also be used to append (add) data from one table to another. Both tables must have the same structure. This operation is achieved by entering the table to which we want to copy the data of the current table and choosing Data only.

For example, we would want to append data when book data comes from various sources (various publishers), is stored in more than one table, and we want to aggregate all the data to one place. For MyISAM, a similar result can be obtained by using the MERGE storage engine (which is a collection of identical MyISAM tables) but if the table is InnoDB, we need to rely on phpMyAdmin's Copy feature.

Multi-Table Operations

In the Database view, there is a checkbox next to each table name and a drop-down menu under the table list. This enables us to quickly choose some tables and perform an operation on all those tables at once. Here we select the book-copy and the book tables, and choose the Check operation for these tables.

We could also quickly select or deselect all the checkboxes with Check All/Uncheck All.
Table and Database Operations

Repairing an "in use" Table

The multi-table mode is the only method (unless we know the exact SQL query to type) for repairing a corrupted table. Such tables may be shown with the in use flag in the database list. Users seeking help in the support forums for phpMyAdmin often receive this tip from experienced phpMyAdmin users.

Database Operations

The Operations tab in the Database view gives access to a panel that enables us to perform operations on a database taken as a whole.
Chapter 10

Renaming a Database
Starting with phpMyAdmin 2.6.0, a Rename database dialog is available. Although this operation is not directly supported by MySQL, phpMyAdmin does it indirectly by creating a new database, renaming each table (thus sending it to the new database), and dropping the original database.

Copying a Database
Since phpMyAdmin 2.6.1, it is possible to do a complete copy of a database, even if MySQL itself does not support this operation natively.

Summary
In this chapter, we covered the operations we can perform on whole tables or databases. We also took a look at table maintenance operations for table repair and optimization, changing various table attributes, table movements, including renaming and moving to another database, and multi-table operations.
Where to buy this book

You can buy Mastering phpMyAdmin 2.11 for Effective MySQL Management from the Packt Publishing website:

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.