# 7

# Advanced Data Access and Display

*Information about how users are interacting with an educational program can provide insight into how well the program is serving the needs of its students. Monitoring things such as, but not limited to, student interaction, attendance, and assessed performance will help educators and educational administrators to make decisions about the curriculum and its execution. The ability, for example, to determine how many times a student interacts with the Moodle site, may provide insight into the student's level of motivation and interest. Additionally, the ability to monitor a student's performance as determined by assessment systems, automated or teacher-generated, may enable educational administrators to better assist students in need of guidance and to adjust the curriculum to meet the needs of those who would benefit from more academic challenges.*

In this chapter, we will explore the installation and use of several tools that allow for more advanced methods of accessing data generated and used by your Moodle site. We will cover the following in this chapter:

- Installation and use of the Enhanced User Administration block
- Installation and use of the PHPMyAdmin database admin plugin
- Installation and basic use of the Custom SQL query tool
- Installation and basic setup of the embedded Xataface database application

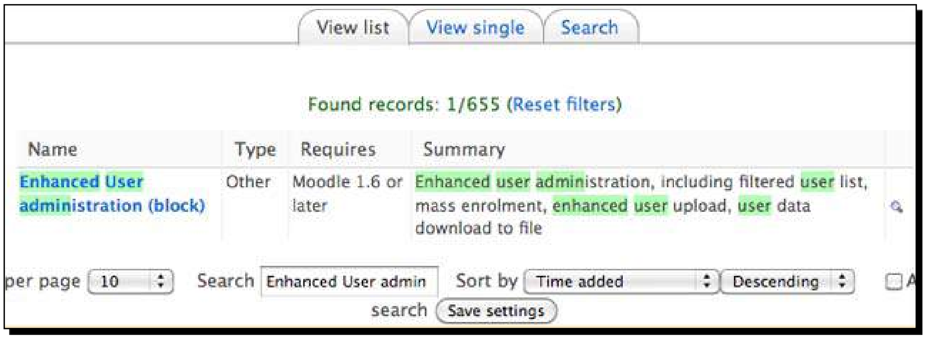
# The Enhanced User Administration block

The Enhanced User Administration block was designed to be used by users who have been granted admin level access to a Moodle site. This tool allows a user to perform various actions including filtering, enrolling, deleting, and others, on his or her accounts on the site. It was originally contributed to the Moodle community to fill a gap that existed in the Moodle standard package. The most recent versions of Moodle 1.9 have however, added a very similar capability, called **Bulk user actions**, that is accessible via the **Site Administration block** by clicking on **Users**, **Accounts**, and then **Bulk user actions**. While very similar in design, the **Enhanced User Administration** block and **Bulk user actions** tools are not identical in design or function and it is thus beneficial to have access to both.

## Time for action – installation and use of the Enhanced User Admin block

Installing the Enhanced User Admin block is a process very similar to the installation of other packages, such as themes and modules, that has been previously explained in this book. To install the block follow these steps:

1. Download the block from `Moodle.org` from the Modules and Plugins page, as shown in the following screenshot. Note that there is one other user administration block available from `Moodle.org` titled simply **User Administration**. Use the **Enhanced User Administration (block)** to obtain the functionality that will be presented here.

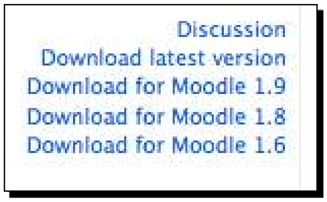


Found records: 1/655 (Reset filters)

| Name | Type | Requires | Summary |
|---|---|---|---|
| Enhanced User administration (block) | Other | Moodle 1.6 or later | Enhanced user administration, including filtered user list, mass enrolment, enhanced user upload, user data download to file |

per page 10 Search Enhanced User admin Sort by Time added Descending
search Save settings

2. Download the block package by first clicking on the **Name** link and then on the **Download latest version** link found in the bottom-right corner of the explanation window, as shown in the following screenshot:



Discussion
Download latest version
Download for Moodle 1.9
Download for Moodle 1.8
Download for Moodle 1.6

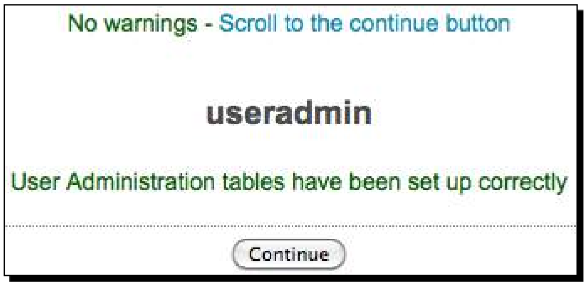
3. Unzip the package that was downloaded to your computer.

4. If the name of the unzipped directory is `useradmin19`, change it to simply `useradmin`.

5. Copy the `useradmin` folder to the `blocks` directory in your Moodle root. This can be found at `/yourmoodleroot/blocks`.

6. Log in to your site as admin and click on the **Notifications** link found in the **Site Administration** block to finish the installation.

7. Click on the **Continue** button, shown in the following screenshot, after confirming the block setup:



No warnings - Scroll to the continue button

useradmin

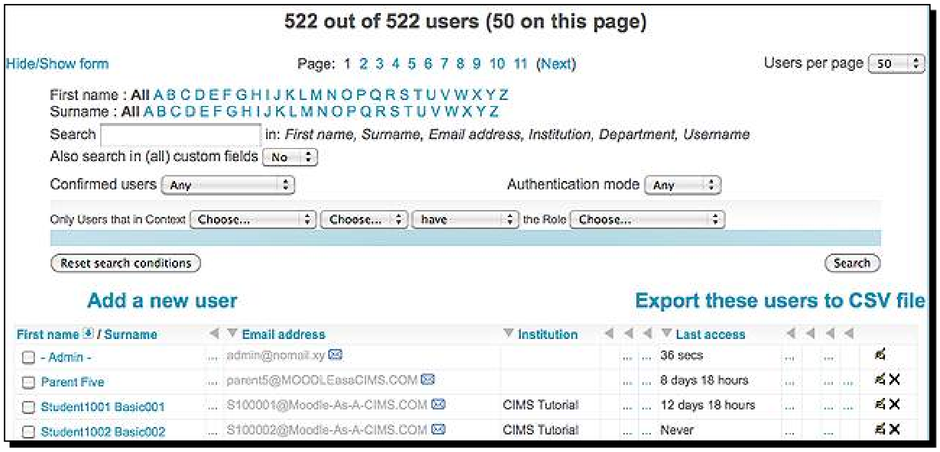
User Administration tables have been set up correctly

8. Go to the front page of your site, turn on editing, and add the **User Administration** block via the **Blocks** pull down menu.

9. Position the block, using the arrows shown with editing turned on, to a location that will be easily accessible. Just above the **Site Administration** block should be a convenient location.

10. Turn editing off and then click on the **Browse list of Users link** to access the block's features.

**11.** You will see the various filter settings at the top of the screen followed by a list of the users in your system, as shown in the following screenshot. Click once on the triangle shown at the top of the user list to display or hide the associated column.



522 out of 522 users (50 on this page)

Hide/Show form — Page: 1 2 3 4 5 6 7 8 9 10 11 (Next) — Users per page 50

First name : All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Surname : All A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Search [ ] in: *First name, Surname, Email address, Institution, Department, Username*

Also search in (all) custom fields: No

Confirmed users: Any — Authentication mode: Any

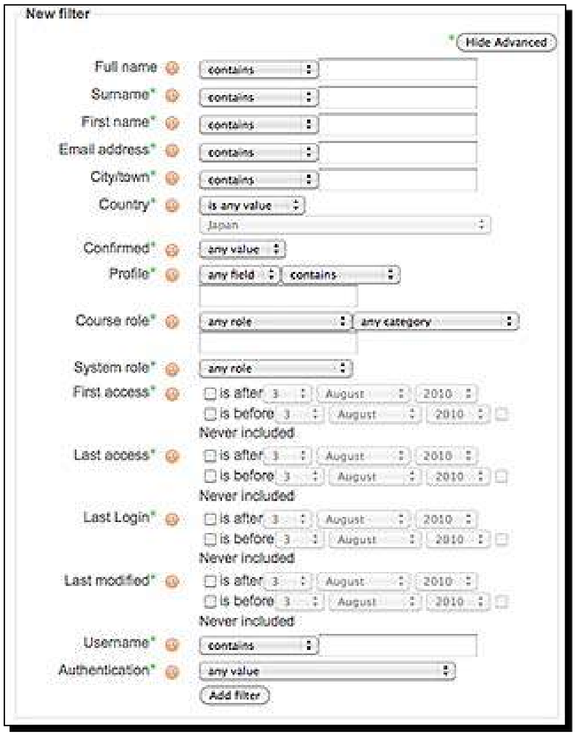
Only Users that in Context: Choose... | Choose... | have | the Role: Choose...

Reset search conditions — Search

Add a new user — Export these users to CSV file

| | First name / Surname | | Email address | Institution | | | Last access | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | - Admin - | ... | admin@nomail.xy | | ... | ... | 36 secs | ... | ... | |
| □ | Parent Five | ... | parent5@MOODLEasaCIMS.COM | | ... | ... | 8 days 18 hours | ... | ... ... | ✕ |
| □ | Student1001 Basic001 | ... | S100001@Moodle-As-A-CIMS.COM | CIMS Tutorial | ... | ... | 12 days 18 hours | ... | ... ... | ✕ |
| □ | Student1002 Basic002 | ... | S100002@Moodle-As-A-CIMS.COM | CIMS Tutorial | ... | ... | Never | ... | ... | ✕ |

**12.** The following screenshot was taken from the window that is viewed via the **Bulk user actions** link, introduced earlier:



New filter

Hide Advanced

Full name: contains

Surname*: contains

First name*: contains

Email address*: contains

City/town*: contains

Country*: is any value — Japan

Confirmed*: any value

Profile*: any field — contains

Course role*: any role — any category

System role*: any role

First access*: □ is after 3 August 2010 / □ is before 3 August 2010 □ Never included

Last access*: □ is after 3 August 2010 / □ is before 3 August 2010 □ Never included

Last Login*: □ is after 3 August 2010 / □ is before 3 August 2010 □ Never included

Last modified*: □ is after 3 August 2010 / □ is before 3 August 2010 □ Never included

Username*: contains

Authentication*: any value

Add filter

13. As mentioned earlier, the two tools are very similar in function. There are however, some key differences that make having access to both tools beneficial. The following are the two key differences in these tools provided in order to demonstrate the functionality offered by the Enhanced User Admin block, hereafter referred to as the UserAdmin block:

    - The UserAdmin block allows for display and sorting (ascending/descending order) of columns from the filter results window. This sorting ability could allow for easy access to users who have accessed the site within the past day, for example. This cannot be performed via the **Bulk user actions** tool. Access information can be displayed via the **Browse a list of users** function found by clicking on **Users**, then **Accounts**, and then **Browse a list of users** from within the **Site Administration** block. Actions cannot however, be performed on the users displayed via the **Browse a list of users** window. Note also that customized user profile fields can also be added to the display window via the block settings that are accessible by clicking on **Modules**, then **Blocks**, and finally **User Administration** from the **Site Administration** block.
    - From the **UserAdmin** search result display window, e-mail addresses can be activated or deactivated by simply clicking on the user's address. The same task can be performed from the **Bulk user actions** tool but must be performed on users returned from a search without knowing the current status of their e-mail. The display window of the **UserAdmin** tool however, displays whether a user's e-mail address is activated or deactivated. If, for example, I need to search for users whose e-mail is deactivated in a course or other specific context, the **UserAdmin** tool would be more appropriate than the **Bulk user actions** tool. If I simply wanted to activate the e-mail address of all users in a given context, the **Bulk user actions** tool would enable me to perform this task more quickly.

## What just happened?

We have just downloaded and installed the Enhanced User Administration (UserAdmin for short) block. A basic comparison of the block with the **Bulk user actions** tool was then made in order to demonstrate the differences in functionality. While these two tools are very similar in function, two explanations of differences supporting the use of the **UserAdmin** block were provided.

## Pop quiz

1. If you want to search for and download a list of users who accessed your Moodle site in the past six hours, which of the tools, just introduced, would you use?
    a. The **UserAdmin** block
    b. The **Bulk user actions** tool
2. If you want to check to see which students in a specific course may not have received communication mailed via the news forum in the course, which of the following would be the most time-efficient method?
    a. Navigate to the course in question, click on the **Participants** link and then check the e-mail status of each participant in the course by clicking each participant and viewing their profile.
    b. Click on **Browse a list of users** from the **User Administration** block, filter for the course in question and check to see which students have e-mail addresses that are not activated from the results display window.
    c. Use the **Bulk user actions** tool to filter for the participants in a course and then send them all a message to check to see if they are receiving e-mails.
3. If you want to activate the e-mail addresses of all the students in a course, which of the following would you use?
    a. The **UserAdmin** block
    b. The **Bulk user actions** tool
    c. The **Participants** display from the course main page

## Have a go hero

While there are arguably more similarities than differences in the UserAdmin block and the Bulk user actions tool, the differences do allow for additional functionality in your Moodle site. Work with both tools and see if you can design a filter or find a search result that is better in one tool than in the other. This activity will help to familiarize you with both tools and better enable you to select the one appropriate to the task you wish to perform.

# Installation and use of PHPMyAdmin

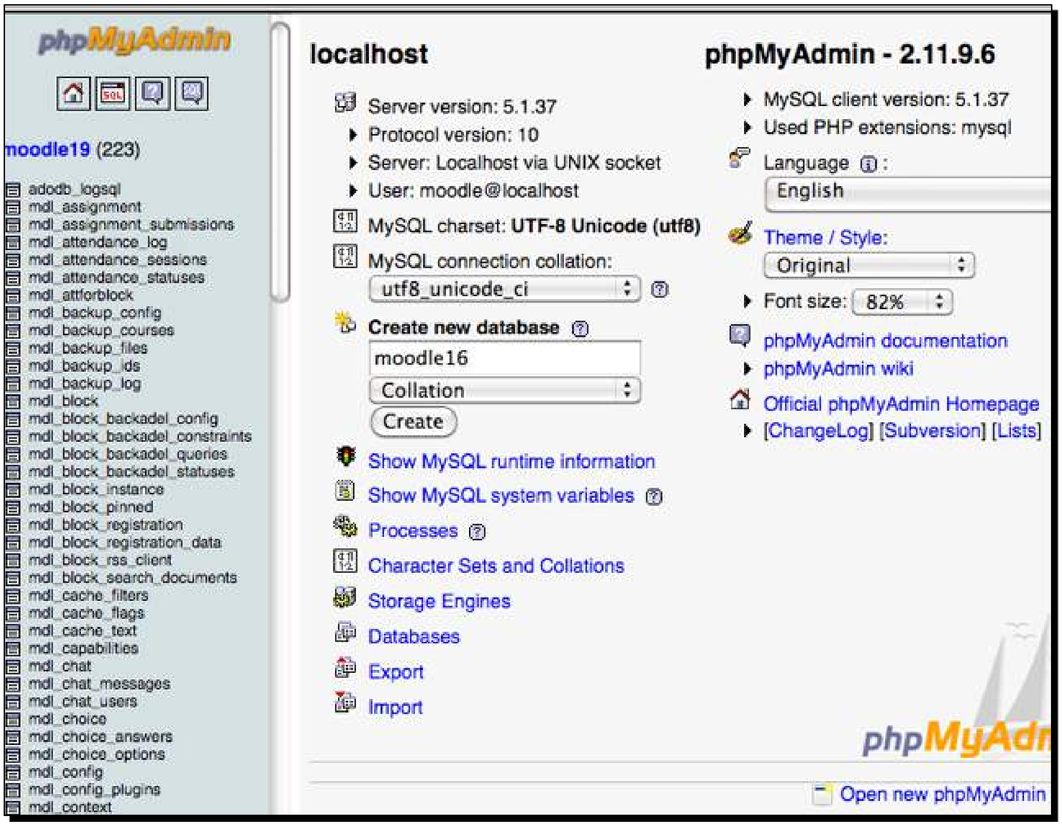
If your Moodle site runs on the MySQL database, as does the site we set up for this book, you may use the PHPMyAdmin plugin to access and administer the database directly. This plugin is a very powerful tool that allows you to access data stored in the Moodle database directly. With power however, also comes the ability to corrupt or completely destroy the Moodle database and all of its contents. It is thus important to use this tool with caution. Many, but not all, of the actions performed via the PHPMyAdmin tool will prompt you for confirmation before being executed and thus give you a last chance before execution. If you aren't sure of yourself, don't do it! Changes made via the PHPMyAdmin tool are often not reversible without considerable effort. Used with caution however, the PHPMyAdmin tool can enable you to very efficiently add to and retrieve information from the Moodle database.

## Time for action – installing and using PHPMyAdmin

To download, install, and start using the PHPMyAdmin tool to give you direct access to the Moodle database, follow these steps:

1. Download the latest version of the PHPMyAdmin tool from the **Modules and Plugins** page on `Moodle.org`.
2. Search for **PHPMyAdmin**, click on the **Name** link and then on the **Download latest version** link found at the bottom-right of the plugin explanation page. (Note, the plugin title on `Moodle.org` is **MySQL Admin**)
3. Unzip the downloaded package and place the directory, titled `mysql`, in the admin directory of your Moodle site located at `/yourmoodleroot/admin`.
4. Log in to your site as admin and click on the **Server** link found in the **Site Administration** block.

**5.** A **Database** link should appear at the bottom of the list of items under the **Server** category. Click on the **Database** link to access the PHPMyAdmin tool, as shown in the following screenshot (Note, if you were already logged in as admin when you installed the package, you may need to refresh your browser window to get the **Database** link to appear).



moodle19 (223)

adodb_logsql
mdl_assignment
mdl_assignment_submissions
mdl_attendance_log
mdl_attendance_sessions
mdl_attendance_statuses
mdl_attforblock
mdl_backup_config
mdl_backup_courses
mdl_backup_files
mdl_backup_ids
mdl_backup_log
mdl_block
mdl_block_backadel_config
mdl_block_backadel_constraints
mdl_block_backadel_queries
mdl_block_backadel_statuses
mdl_block_instance
mdl_block_pinned
mdl_block_registration
mdl_block_registration_data
mdl_block_rss_client
mdl_block_search_documents
mdl_cache_filters
mdl_cache_flags
mdl_cache_text
mdl_capabilities
mdl_chat
mdl_chat_messages
mdl_chat_users
mdl_choice
mdl_choice_answers
mdl_choice_options
mdl_config
mdl_config_plugins
mdl_context

Server version: 5.1.37
- Protocol version: 10
- Server: Localhost via UNIX socket
- User: moodle@localhost

MySQL charset: **UTF-8 Unicode (utf8)**

MySQL connection collation: utf8_unicode_ci

**Create new database**
moodle16
Collation

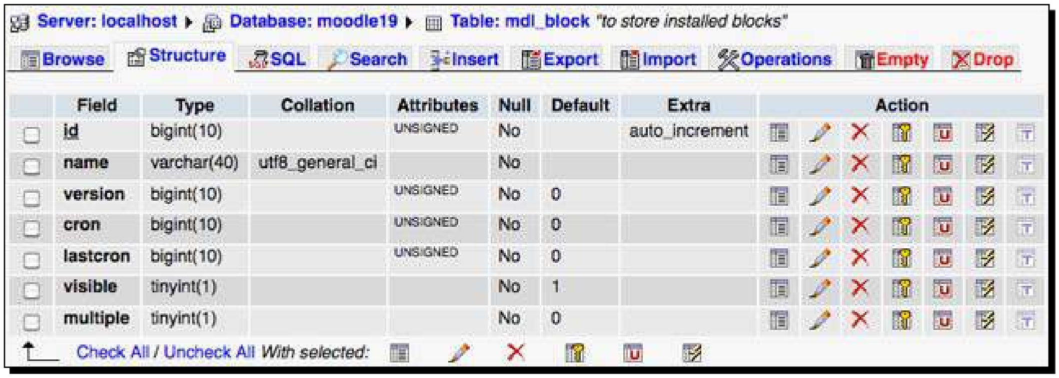
- MySQL client version: 5.1.37
- Used PHP extensions: mysql

Language: English

Theme / Style: Original

- Font size: 82%

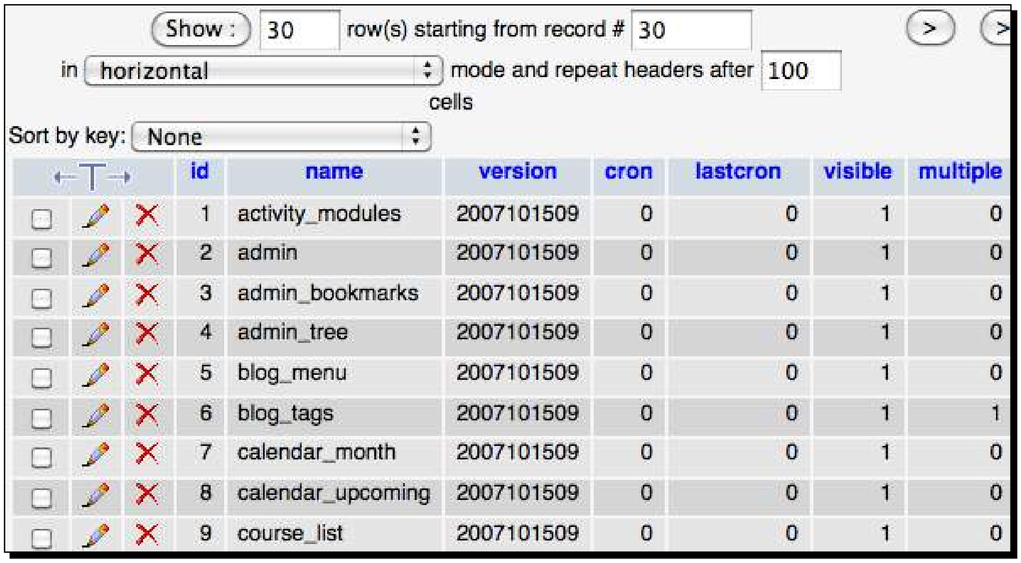
**6.** Each of the lines shown on the left side of the window represents a table in the Moodle database. The database that our example Moodle installation is running on contains 223 tables, as shown in parentheses at the top-left of the screen next to the database name in the previous screenshot. Click on one of the tables, `mdl_block`, for example, to view the **Structure** window for that table, as shown in the following screenshot:


Server: localhost ▸ Database: moodle19 ▸ Table: mdl_block *"to store installed blocks"*
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop

| | Field | Type | Collation | Attributes | Null | Default | Extra | Action |
|---|---|---|---|---|---|---|---|---|
| □ | id | bigint(10) | | UNSIGNED | No | | auto_increment | |
| □ | name | varchar(40) | utf8_general_ci | | No | | | |
| □ | version | bigint(10) | | UNSIGNED | No | 0 | | |
| □ | cron | bigint(10) | | UNSIGNED | No | 0 | | |
| □ | lastcron | bigint(10) | | UNSIGNED | No | 0 | | |
| □ | visible | tinyint(1) | | | No | 1 | | |
| □ | multiple | tinyint(1) | | | No | 0 | | |

Check All / Uncheck All *With selected:*

***7.*** Now click on the **Browse** link, found at the top-left of the screen, to view a window that displays the contents of the table, as shown in the following screenshot:


Show : 30 row(s) starting from record # 30
in horizontal mode and repeat headers after 100 cells
Sort by key: None

| | | | id | name | version | cron | lastcron | visible | multiple |
|---|---|---|---|---|---|---|---|---|---|
| □ | | | 1 | activity_modules | 2007101509 | 0 | 0 | 1 | 0 |
| □ | | | 2 | admin | 2007101509 | 0 | 0 | 1 | 0 |
| □ | | | 3 | admin_bookmarks | 2007101509 | 0 | 0 | 1 | 0 |
| □ | | | 4 | admin_tree | 2007101509 | 0 | 0 | 1 | 0 |
| □ | | | 5 | blog_menu | 2007101509 | 0 | 0 | 1 | 0 |
| □ | | | 6 | blog_tags | 2007101509 | 0 | 0 | 1 | 1 |
| □ | | | 7 | calendar_month | 2007101509 | 0 | 0 | 1 | 0 |
| □ | | | 8 | calendar_upcoming | 2007101509 | 0 | 0 | 1 | 0 |
| □ | | | 9 | course_list | 2007101509 | 0 | 0 | 1 | 0 |

## What just happened?

We have just downloaded and installed a database access and administration tool called PHPMyAdmin. We looked at the database via the PHPMyAdmin tool and viewed the contents of one table in our Moodle database. Interaction between the user (student, teacher, or administrator), and the database is normally facilitated by Moodle. For some specific administrative tasks however, using a tool that allows for access to the database directly, can enable an administrator to perform tasks in bulk much more efficiently than they could be performed through the Moodle site.

Until you become familiar with the structure of the Moodle database and are comfortable navigating through it using PHPMyAdmin, this tool may appear to be less than useful. In the next chapter of this book however, we will introduce a method for inserting test scores and other educational program-related data into the database that utilizes the PHPMyAdmin tool. While these tasks are possible via other methods, the use of PHPMyAdmin makes inserting data into the database in bulk a very fast, straightforward and relatively simple process. You will certainly begin to appreciate all that can be done with PHPMyAdmin as you start to use it for these types of tasks.

## Have a go hero

You may be thinking that PHPMyAdmin looks like a very daunting tool that you'd rather not work with. Try the following so you don't have to wait until *Chapter 8*, *Setting Up a Mini SIS* to start to see how useful PHPMyAdmin can be. Scroll down the list of tables and find the table titled `mdl_user`. Browse the contents of this table and then browse through some other tables such as `mdl_attendance_log` and know that many of the tables are related to each other via one or more ID numbers. Think about how an educational administrator might be able to use data stored in the Moodle database if he or she could access and view it in an easy-to-understand format.

# Installation and basic use of the Custom SQL query tool

The ability to create ad hoc queries that retrieve specific data from the Moodle database can enable you, as an educational administrator, to create monitoring tools that inform you and others on things such as the results of a test, attendance figures, and any other plethora of scenarios that you may dream up. Use of this tool will require basic SQL knowledge at the least and more advanced skills for sophisticated queries. There are many resources on the web that will help to get you started with SQL. A good place to start is `www.sql.org`.

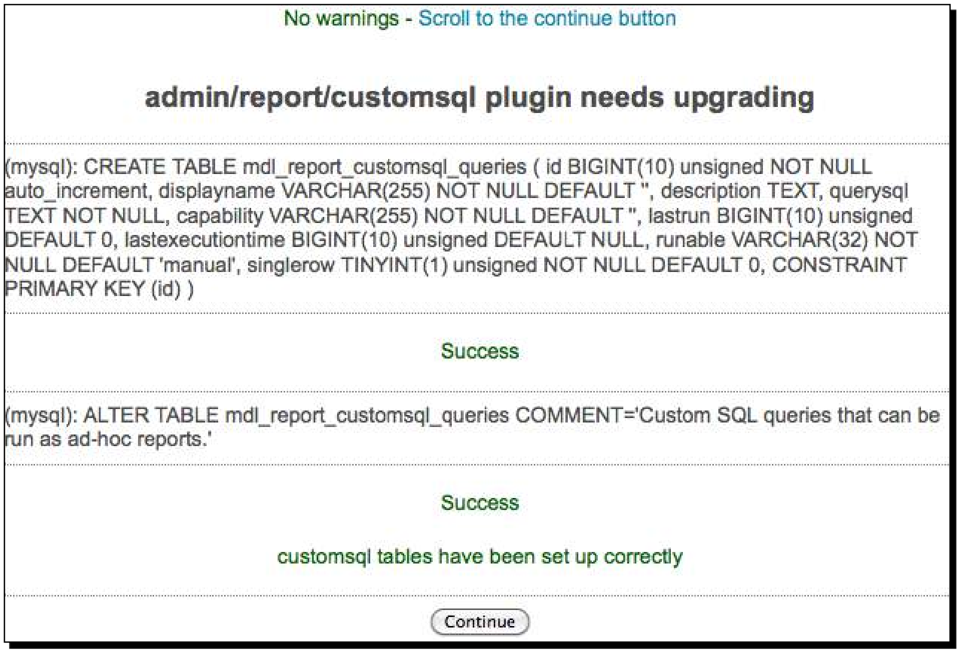
## Time for action – install and experiment with the Custom SQL query tool

Installation of the Custom SQL query tool is a straightforward process that resembles the process for installing blocks, modules, and other plugins. To install the tool follow these steps:

1. From the `Moodle.org`, **Modules and Plugins** page, search for **Custom SQL queries**.
2. Click on the **Custom SQL queries** name link found in the search results and then on the **Download for 1.9** link found at the bottom-right corner of the explanation window.

3. Unzip the `customsql.zip` package and place the resulting `customsql` directory in the admin reports directory found in your Moodle directory at `/yourmoodledirectory/admin/reports`.

4. Click on the **Notifications** link found in the **Site Administration** block to finish the installation process. You will see a screen that confirms that the customsql plugin has been set up correctly, as shown in the following screenshot:



No warnings - Scroll to the continue button

admin/report/customsql plugin needs upgrading

```
(mysql): CREATE TABLE mdl_report_customsql_queries ( id BIGINT(10) unsigned NOT NULL auto_increment, displayname VARCHAR(255) NOT NULL DEFAULT '', description TEXT, querysql TEXT NOT NULL, capability VARCHAR(255) NOT NULL DEFAULT '', lastrun BIGINT(10) unsigned DEFAULT 0, lastexecutiontime BIGINT(10) unsigned DEFAULT NULL, runable VARCHAR(32) NOT NULL DEFAULT 'manual', singlerow TINYINT(1) unsigned NOT NULL DEFAULT 0, CONSTRAINT PRIMARY KEY (id) )
```

Success

```
(mysql): ALTER TABLE mdl_report_customsql_queries COMMENT='Custom SQL queries that can be run as ad-hoc reports.'
```

Success

customsql tables have been set up correctly

5. Click on the **Continue** button to finish the installation and then navigate to the front page of your Moodle site.

6. Click on the **Reports** link found in the **Site Administration** block and then on the **Ad-hoc database** queries link to access the Custom SQL queries tool.

## What just happened?

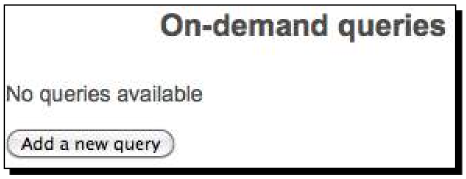
We have just downloaded and installed the Custom SQL queries administrator tool. This tool will allow us to construct ad hoc queries to meet the various data gathering needs that exist in our educational institution.

# SQL queries

As was mentioned earlier, writing SQL queries requires some basic knowledge of SQL and understanding of its logic. We will walk through the creation of one fairly simple query to demonstrate the power that is presented by this block.

## Time for action – creating a sample query

The first time you access the Custom SQL queries tool via the **Ad-hoc database queries** link, you will see a screen that indicates there are **No queries available**, as shown in the following screenshot:

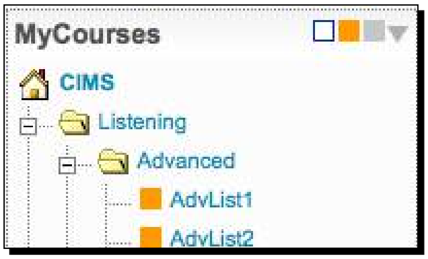


On-demand queries

No queries available

Once you create one or more SQL queries, the queries will appear in this window. To demonstrate the usefulness of this tool, we will set up a simple sample query. Our query is going to be designed to pull final grades from courses, so we will first need to navigate to one of our courses and enter several grades to ensure that we have data to pull from the database. Follow these steps:

1. Navigate to the **Advanced Listening 1** course on our test site by clicking on the **AdvList1** link in the **MyCourses** block, as shown in the following screenshot:

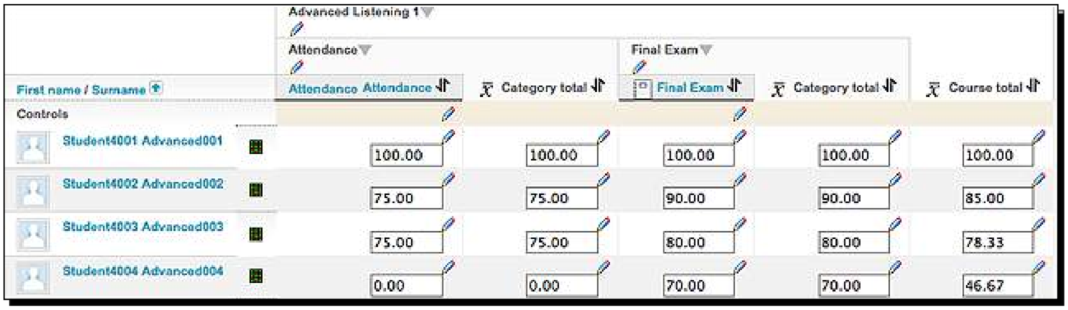


2. There are two graded items here, **Attendance** and **Final Exam**, from our course template imported from *Chapter 4*, *Incorporating Educational Standards*, so we already have some items that can be graded. You could enter attendance data and final exam scores by clicking on the item link (**Attendance** and **Final Exam**) found in the course and entering scores but we will take an easier approach for this activity. Click on the **Grades** link found in the **Administration** block on the left side of the screen.

**3.** You will now see the **Grader report** screen. Use the drop-down menu found in the top left-hand side of the window if you are in a different view.

**4.** Push the **Turn editing on** button found at the top-right of the screen.

**5.** Enter some scores for a few students, as shown in the following screenshot. (Note, the **Course total** figures are calculated by the weighting scheme setup in *Chapter 4*, *Incorporating Educational Standards*.)



|  | Advanced Listening 1 |  |  |  |  |
|---|---|---|---|---|---|
|  | Attendance |  | Final Exam |  |  |
| First name / Surname | Attendance Attendance | Category total | Final Exam | Category total | Course total |
| Controls |  |  |  |  |  |
| Student4001 Advanced001 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| Student4002 Advanced002 | 75.00 | 75.00 | 90.00 | 90.00 | 85.00 |
| Student4003 Advanced003 | 75.00 | 75.00 | 80.00 | 80.00 | 78.33 |
| Student4004 Advanced004 | 0.00 | 0.00 | 70.00 | 70.00 | 46.67 |

**6.** Click on the **Update** button found at the bottom of the screen.

**7.** Navigate back to the front page of your site.

**8.** Now that we have entered some grades into the system, we are ready to build and use an SQL query via the Ad-hoc database query tool. First, click on the **Ad-hoc database queries** link found by clicking on the **Reports** link located in the **Site Administration** block.

**9.** Click on the **Add a new query** button.

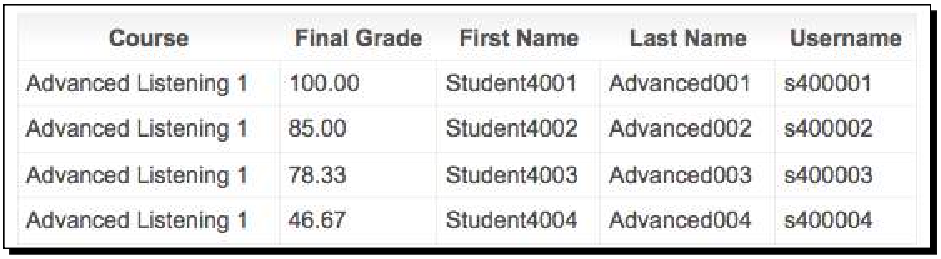
**10.** The **Editing an ad-hoc database query** screen will appear. Enter a **Query name** and **description** such as **Course Grades** and **Final scores from courses**.

**11.** In the Query SQL window, enter the following SQL code:

```
SELECT
prefix_course.fullname AS «Course»,
ROUND(prefix_grade_grades.finalgrade, 2) AS «Final Grade»,
prefix_user.firstname AS «First Name»,
prefix_user.lastname AS «Last Name»,
prefix_user.username AS «Username»
FROM
prefix_grade_grades
INNER JOIN prefix_user ON prefix_grade_grades.userid = prefix_
user.id
```

```
INNER JOIN prefix_grade_items ON prefix_grade_grades.itemid =
prefix_grade_items.id
INNER JOIN prefix_course ON prefix_grade_items.courseid = prefix_
course.id
WHERE prefix_grade_items.id = prefix_grade_grades.itemid
AND (prefix_grade_items.courseid = prefix_course.id)
AND (prefix_grade_grades.finalgrade > 0)
AND (prefix_user.deleted = 0)
AND prefix_grade_items.itemtype = ‹course›
ORDER BY prefix_course.id, prefix_user.id
```

12. Leave the **Who can access this query**, **Run**, and **Type of result** settings at their default values and click on the **Save changes** button located at the bottom of the page. The query will be executed and the results will be displayed, as shown in the following screenshot:



| Course | Final Grade | First Name | Last Name | Username |
|---|---|---|---|---|
| Advanced Listening 1 | 100.00 | Student4001 | Advanced001 | s400001 |
| Advanced Listening 1 | 85.00 | Student4002 | Advanced002 | s400002 |
| Advanced Listening 1 | 78.33 | Student4003 | Advanced003 | s400003 |
| Advanced Listening 1 | 46.67 | Student4004 | Advanced004 | s400004 |

## What just happened?

We just successfully downloaded and installed the Custom SQL query tool and set up one SQL query designed to pull final grades directly from the table used by Moodle to store grades. This tool can be used to extract virtually any data from the Moodle database that you wish to access. If you have a systems engineer or technical support personnel on staff, you will likely be able to get help in designing advanced queries to access your Moodle database. If access to such personnel is limited you may need to search the web for helpful resources, purchase some introductory SQL manuals, or hire an outside contractor to help you get your system set up. Prior to doing so however, you should spend some time getting yourself familiarized with the Moodle database via browsing the tables and their fields with a tool such as the PHPMyAdmin plugin we installed earlier in this chapter. An additional place to start when experimenting with SQL queries is the Custom SQL query tool documentation page on Moodle.org found at `http://docs.moodle.org/en/Custom_SQL_queries_report`.

## Have a go hero

We have demonstrated just one possible type of query that can be constructed with the Custom SQL query tool. Try and see if you can think of other queries that would provide valuable information to you and your institution as you position your Moodle site as a CIMS. Visit the Custom SQL queries report page at the Moodle Docs site (`http://docs.moodle.org`) for ideas and, once you've designed that perfect query, post it in the Docs for others to use!

Also, as this book goes to press, a new database report generating tool, in the form of a block, has been contributed to the Moodle community. The tool can be found and downloaded from the Modules and Plugins database on the `Moodle.org` website. Search for it using the string "`Block: Configurable Reports`". This tool is evolving quickly and promises to be a tool that will allow for customized reports that can be generated via a browser interface with little to no knowledge of SQL. It has been contributed and is being maintained by *Juan Leyva* from Madrid, Spain. Download it and give it a try and compliment Juan on his fabulous contribution!

# Installation and basic setup of the embedded Xataface database application

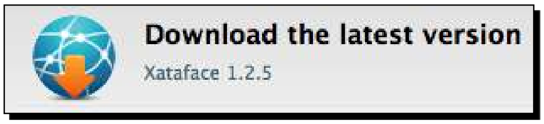
In the previous two sections, we've seen that there is an enormous amount of data stored in the Moodle database and that some of it can be of value to educational program administrators. The limitation of both of the tools previously introduced is that both require some technical skill to use and can introduce serious vulnerabilities in terms of the types of access that they grant to the database. This is where a tool that allows educational administrators access to the Moodle database on limited terms can be very useful. The tool that we will introduce here and expand upon in *Chapter 8*, *Setting Up a Mini SIS*, is a database access tool called Xataface. Xataface is open source, just like Moodle, and has a relatively smooth learning curve.

## Time for action – installation and basic setup of Xataface

Xataface does introduce a method for users to access the Moodle database directly, so we'll want to be very careful in how we install and set it up. It is highly recommended that you first set up Xataface in a test environment such as the MAMP server we have set up for this book. Follow these steps to get your Xataface tool downloaded, installed, and set up:

1. Navigate to the Xataface website found at: `http://xataface.com`.

2. Click on the **Download the latest version** link found on the front page of the site, as shown in the following screenshot:

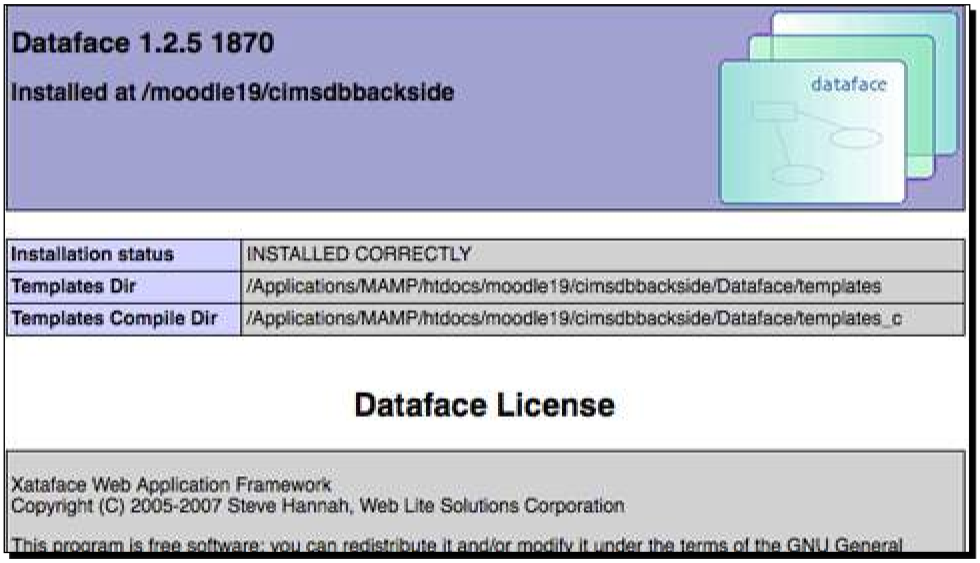


3. The link will redirect you to a SourceForge page from which the latest version can be downloaded. The latest version will be at the top of the list and packaged both as `tar.gz` and `zip` packages. Download either of the compressed packages.

4. Expand the contents of the package and rename it to `Xataface`.

5. Create a new directory in your Moodle root directory called `CIMS` and place the `Xataface` directory and all of its contents inside the `CIMS` directory.

6. With your browser, navigate to the file titled `dataface_info.php` found in the directory that was just placed in the Moodle directory. Using the name we provided in step 4 earlier would result in `yourmoodlesite/cims/xataface/dataface_info.php`. You should see the following screen, which will mean that your installation is functioning properly:



**Dataface 1.2.5 1870**

**Installed at /moodle19/cimsdbbackside**

| | |
|---|---|
| **Installation status** | INSTALLED CORRECTLY |
| **Templates Dir** | /Applications/MAMP/htdocs/moodle19/cimsdbbackside/Dataface/templates |
| **Templates Compile Dir** | /Applications/MAMP/htdocs/moodle19/cimsdbbackside/Dataface/templates_c |

**Dataface License**

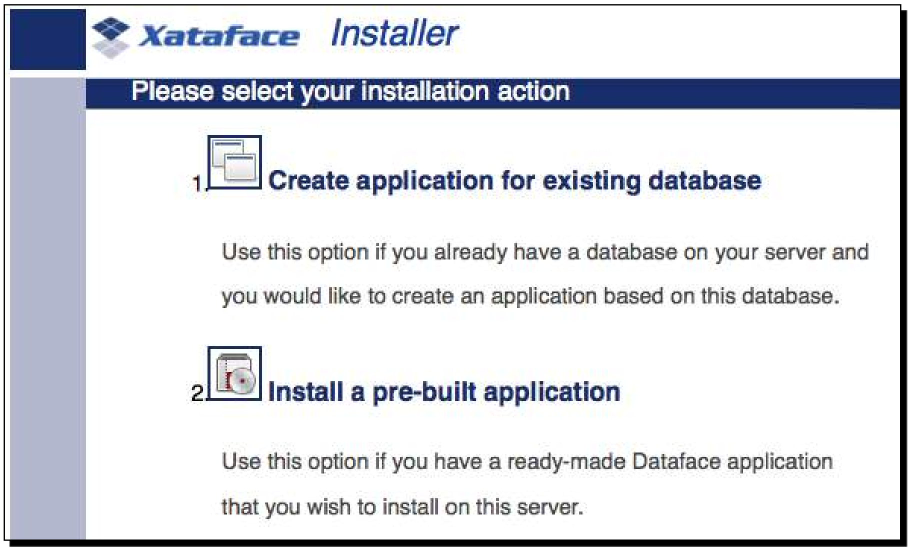
Xataface Web Application Framework
Copyright (C) 2005-2007 Steve Hannah, Web Lite Solutions Corporation

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General

7. Once you have confirmed that the application is installed correctly, direct your browser to the following address: `yourmoodlesite/cims/xataface/installer.php`.

8. You will be prompted to enter the username and password for your database. If you have set up your development site as you worked through this book, both your username and password will be `moodle`. To confirm this information, look for the `config.php` file found in your `Moodle` directory and open it with your text editor. The user name and password are listed as: **dbuser** and **dbpass**.

9. Once you have entered the username and password and clicked enter, you will see a screen that presents you with two options, as shown in the following screenshot. Choose the first option, **Create application for existing database**, as we want to use the Moodle database.



Xataface Installer

Please select your installation action

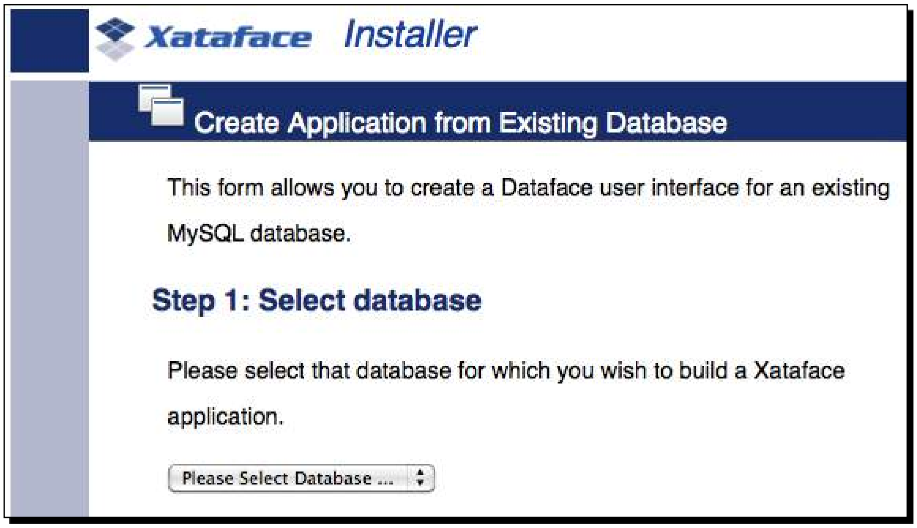
1. Create application for existing database

Use this option if you already have a database on your server and you would like to create an application based on this database.

2. Install a pre-built application

Use this option if you have a ready-made Dataface application that you wish to install on this server.

10. The next screen, as shown in the following screenshot, will prompt you to select a database using a drop-down menu. Pick the item that contains the word Moodle, which should be **moodle19**.



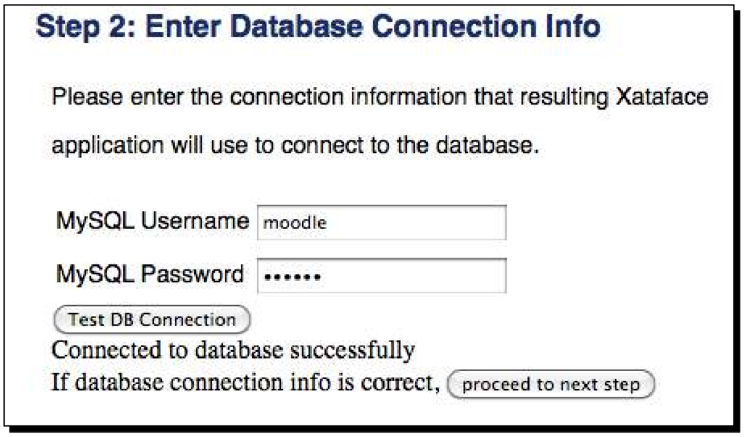
Create Application from Existing Database

This form allows you to create a Dataface user interface for an existing MySQL database.

Step 1: Select database

Please select that database for which you wish to build a Xataface application.

11. After selecting the database, you will be prompted to enter connection information, which consists of the username and password. These fields will already be populated with the username and password that you used to access the installer, so you should only need to click on the **Test DB Connection** button that will result in a **Connected to database successfully** message, on this screen before clicking on **proceed to next step**. The following screenshot is of the step 2 portion of the installation:



Step 2: Enter Database Connection Info

Please enter the connection information that resulting Xataface application will use to connect to the database.

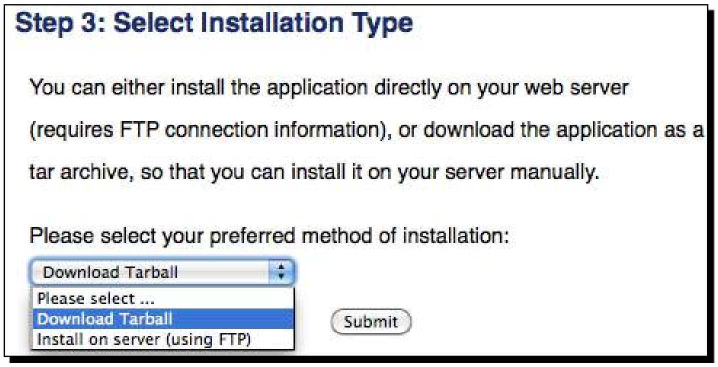
MySQL Username moodle

MySQL Password ••••••

Connected to database successfully

If database connection info is correct,

12. Clicking on **proceed to next step** will present Step 3, as shown in the following screenshot. Use the pull-down menu to select **Download Tarball** as we will be installing this in our local instance of Moodle. Click on the **Submit** button to initiate the download.



Step 3: Select Installation Type

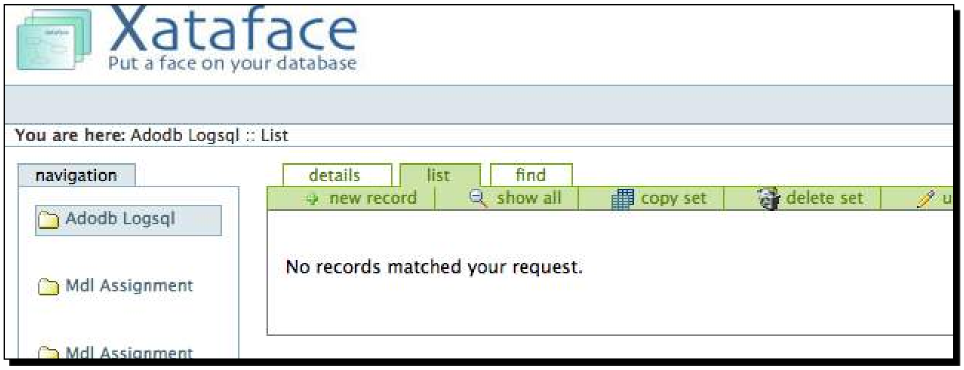
You can either install the application directly on your web server (requires FTP connection information), or download the application as a tar archive, so that you can install it on your server manually.

Please select your preferred method of installation:

13. Expand the package that was downloaded and you will find a file titled `conf.ini` and one titled `index.php`. Place these two files inside the `CIMS` directory but not inside the `Xataface` directory.

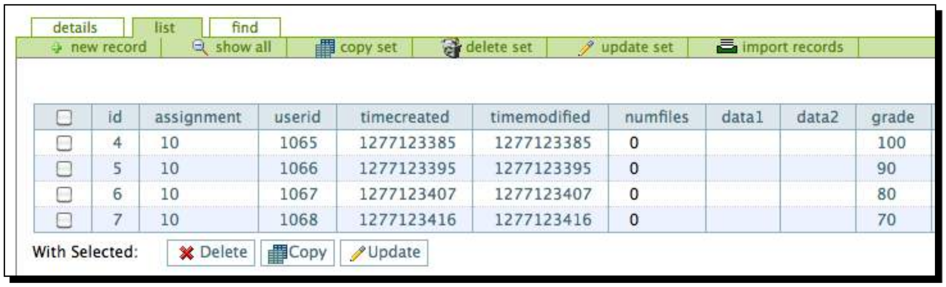
14. Now, point your browser to `yourmoodlesite/cims/index.php` to access your Xataface application.

15. You will see a screen that resembles the following screenshot with access to all of the tables in your Moodle database. This would, of course, be unacceptable in a production environment, first because there is no authentication required for accessing the Xataface program at this point, and second because access is being granted to all tables in the Moodle database, which is unnecessary and unwanted. We'll work on basic setup next and will walk through further customization in the next chapter.



No records matched your request.

16. To get an idea of how Xataface displays table information, click on the **Mdl Assignment Submissions** link, shown on the left side of the page in the **navigation** menu, to view the contents of the table. The following screenshot is what you will see if you entered the grades for the SQL Query tool, explained earlier in this chapter.



| □ | id | assignment | userid | timecreated | timemodified | numfiles | data1 | data2 | grade |
|---|---|---|---|---|---|---|---|---|---|
| □ | 4 | 10 | 1065 | 1277123385 | 1277123385 | 0 | | | 100 |
| □ | 5 | 10 | 1066 | 1277123395 | 1277123395 | 0 | | | 90 |
| □ | 6 | 10 | 1067 | 1277123407 | 1277123407 | 0 | | | 80 |
| □ | 7 | 10 | 1068 | 1277123416 | 1277123416 | 0 | | | 70 |

## *What just happened?*

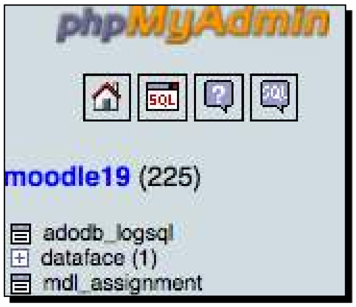
We have just successfully downloaded and installed the Xataface database access application. We then walked through the setup process in order to create an application that uses the Moodle database. We tested Xataface's functionality by clicking on a table link from the **navigation** menu found on the left side of the screen, to view the contents of the table. This application can provide an ideal backside access portal to the Moodle database that will enable educational administrators to field data from and enter information into the Moodle database.

## Time for action – setup of our Xataface application

As we mentioned earlier, the default installation of Xataface is not appropriate as a backside portal for Moodle due to the fact that it grants universal and unlimited access to all of the tables in the Moodle database. There are a wealth of tutorials to help assist you customize your Xataface instance on the web in the Wiki area of the Xataface website at `www.xataface.com`. We will walk through a few of those that are considered to be crucial when using this package together with Moodle. We will start by setting up the authentication feature in Xataface in order to limit access to users who have been granted usernames and passwords. The types of access that can be granted to users will also be listed. To enable the authentication function, follow these steps:

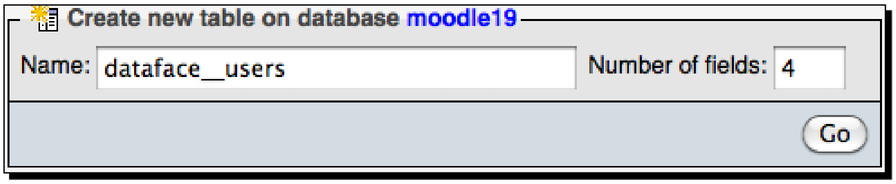
1. Before we set up authentication in the Xataface application, we need to create a table in the database to store user information. This is a perfect chance for us to use the PHPMyAdmin tool, we installed earlier in this chapter. Navigate to the front page of your site, click on the **Server** link located in the **Site Administration** block and then on **Database** to access the database with PHPMyAdmin.

2. Click on the database name link found at the top-left of the screen, as shown in the following screenshot. In our test site case, this will be **moodle19**.



3. Scroll to the bottom of the main screen (not the table list on the left) and enter the following name in the **Name** field window under the **Create new table on database moodle19** heading: `dataface__users`. (Note, use two underbars here to include this table in **dataface** table heading in the table list, as seen in the previous screenshot. Adding this table will result in the number 2 in parenthesis after the **dataface** table title.)

4. Enter a `4` for the **Number of fields** and then click on the **Go** button. The following screenshot is an example of what you should see prior to clicking on the **Go** button:



Name: dataface__users

Number of fields: 4

5. The next window will be a very wide window that will require you to scroll to the right to enter all of the required information. Use the following for the **Field** names, **Type**, and **Length/Values**:

| Field | Type | Length/Values |
|---|---|---|
| UserID | INT | 11 |
| UserName | VARCHAR | 32 |
| Password | VARCHAR | 32 |
| Role | ENUM | 'READ ONLY', 'NO ACCESS', 'ADMIN', 'REVIEWER' |

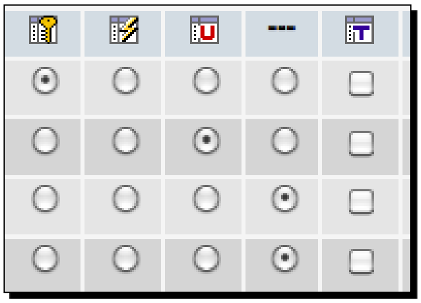
6. Set the **Collation** to **utf8_general_ci** for all fields except the **UserID**.

7. Set the **Role** field to **NULL** but leave the other three **NOT NULL**.

8. The **Default** value for the **Role** field should be **READ ONLY**.

9. Set the **UserID** to **auto_increment** using the **Extra** pull-down menu.

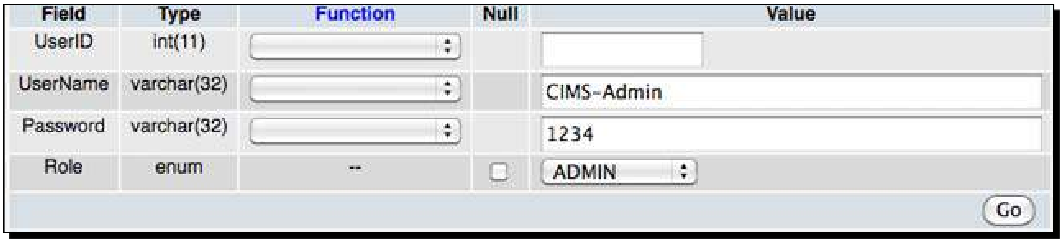
10. Set the **UserID** as the **Primary key** by clicking on the radio button under the key. The **UserName** should be set to unique and the other two fields can be left untouched. The following screenshot is of these radio button settings:



|  |  |  | --- |  |
|---|---|---|---|---|
| ● | ○ | ○ | ○ | □ |
| ○ | ○ | ● | ○ | □ |
| ○ | ○ | ○ | ● | □ |
| ○ | ○ | ○ | ● | □ |

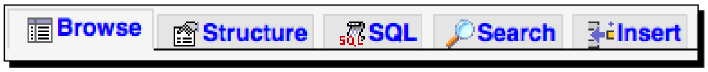
11. After having entered and selected all of the desired settings for the table, click on the **Save** button found on the far right of the screen.

12. The table will be created and from here you will need to enter one record to be used by the site admin. Click on **Insert** found at the top of the screen and enter a **UserName**, **Password**, and select the **ADMIN** role from the pull-down menu. The **UserID** field can be left blank as it will be automatically generated by the database. The following screenshot is an example of how you should fill in the information to create a new record in this table. Click on the **Go** button to enter the new record into the table.

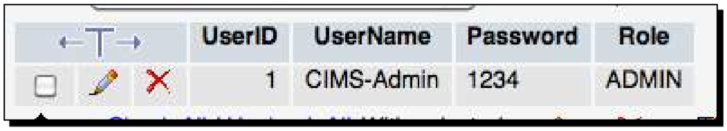


| Field | Type | Function | Null | Value |
|---|---|---|---|---|
| UserID | int(11) |  |  |  |
| UserName | varchar(32) |  |  | CIMS-Admin |
| Password | varchar(32) |  |  | 1234 |
| Role | enum | -- | □ | ADMIN |

**13.** From the next screen, click on the **Browse** link found at the top of the window, as shown in the following screenshot, to verify the new record in the table:



**14.** The record will appear in the table, as shown in the following screenshot. Next we will go to the Xataface application and activate the authentication feature and point it to this table in the database.



|  | UserID | UserName | Password | Role |
|---|---|---|---|---|
| □ | 1 | CIMS-Admin | 1234 | ADMIN |

**15.** Now let's go to the Xataface application and enable the authentication feature and tell Xataface where to find the user information. Open the `conf.ini` file found in your Xataface application directory with your text editor. The directory containing the `conf.ini` file will be `CIMS`, if you have followed the instructions thus far.

**16.** In the `conf.ini` file, below the `[_database]` information, enter the following lines and then save the file:

```
[_auth]
users_table = dataface__users
username_column = UserName
password_column = Password

[allowed_tables]
Users = dataface__users
```

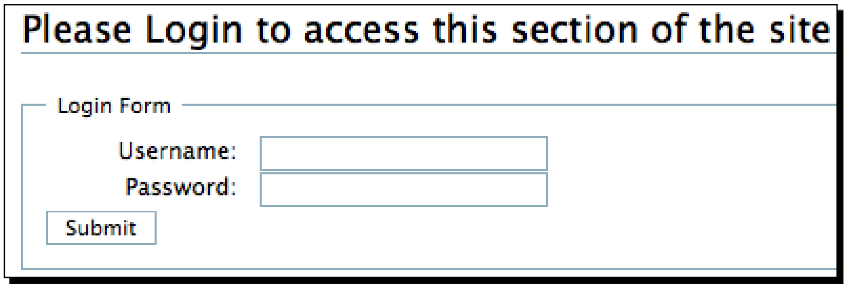
**17.** This code activates the authentication plugin and tells Xataface what table the user account information is stored in. Additionally, the `Users = dataface__Users` record was added under the `[allowed_tables]` heading because, by default, Xataface disallows all tables that begin with `dataface__`.

**18.** In your `CIMS` directory, create a new directory titled `conf`. Using your text editor, create a file inside the `conf` directory and type in the following code:

```
<?php class conf_ApplicationDelegate {      function
getPermissions(&$record){// $record is a Dataface_Record object
$auth =& Dataface_AuthenticationTool::getInstance();
$user =& $auth->getLoggedInUser();
if ( $user ) return Dataface_PermissionsTool::ALL();
else return Dataface_PermissionsTool::NO_ACCESS();      } }
```

19. Save the file as `ApplicationDelegate.php` inside the `conf` directory. This code tells Xataface to require a login of anyone who navigates to the application.

20. Now access your Xataface application by navigating to `/yourmoodlesite/cims/` with your web browser. You will see a login window, as shown in the following screenshot, and notice that while the tables are all still listed, you cannot view the contents of any of them. To log in to the application, use the username and password we entered into the users table earlier: `CIMS-Admin` and `1234`.



Please Login to access this section of the site

Login Form

## *What just happened?*

We just created a new table in our Moodle database to house the user information for our Xataface application. We then populated the table with one user account, our admin account, and then changed the Xataface settings to require a login in all users and told Xataface where to look for user account information. Finally, we logged in to the application with the admin account we created to make sure everything functions as desired. Next, we'll continue to refine the appearance of our application by limiting the tables that are displayed from the database.

## Time for action – limiting database table access

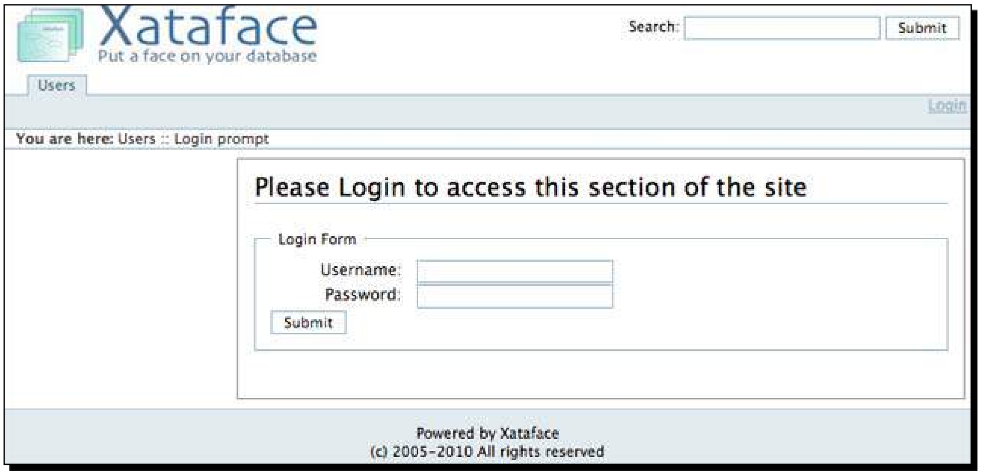
In order to clean up the appearance and to heighten security, we will remove the table navigation display from the left side of the screen and change settings that control access to the tables in the Moodle database. Follow these steps:

1. Once again, open the `conf.ini` file found in the `CIMS` directory located inside your Moodle directory. The `[_tables]` list, that you will see inside this file, is what controls the tables being displayed in the **navigation** column. This `[_tables]` heading only controls the display however, and even if a table is removed from this list, the table can still be accessed by typing the table name into the browser window. We will use the following two directives to either display a table or prevent access to it: `[_tables]` and `[_disallowed_tables]`.

2. The `[_tables]` directive formatting requires the table be identified first and then given a title as in `mdl_user = "Users"`. Add the tables that you wish to allow to this section. For our example, we will only add `Mdl_user` for now so we will add the line, `mdl_user =  "Users"` to our `[_tables]` list.

3. The `[_disallowed_tables]` directive requires a name be specified first followed by an equals sign and then the table name as in `Mdl_user_info_category=mdl_user_info_category`. The `[_tables]` list however, is formatted with the table name first followed by an equals sign and title enclosed in quotation marks, including spaces between words. Inverting the order of these and removing the quotation marks and spaces is the easiest method of creating the `[_disallowed_tables]` list. The following calculation is one way to invert the table title and name after copying and pasting the table list to a spreadsheet: `=SUBSTITUTE(CONCATENATE((MID(A1,(FIND("=",A1,1)+3),(LEN(A1)-(FIND("=",A1,1)+3)))),"=",LEFT(A1,(FIND("=",A1,1)-2)))," ","")` Download the sample `conf.ini` file from the Packt website for reference if you have difficulties or would like to view a completed sample. A sample Excel spreadsheet with this conversion calculation is also available.

4. Once you have completed the process of adding your table to be listed under the `[_tables]` heading and including all of the tables you want to deny access to under the `[_disallowed_tables]` heading, you can save the file and navigate to `yourmoodlesite.com/cims` to view your Xataface application. The first screen you will see should look virtually identical to the following screenshot:


Xataface
Put a face on your database
Search:
Submit
Users
Login
You are here: Users :: Login prompt

Please Login to access this section of the site

Login Form
Username:
Password:
Submit
Powered by Xataface
(c) 2005-2010 All rights reserved

5. Note the **Users** tab at the top of the window in the previous screenshot. This is the only table that we have chosen to display via the [_tables] heading in the conf.ini file and is thus the only table that is accessible. And, of course, it is only accessible after logging in to the application. Log in with the admin **Username** and **Password** we created earlier: CIMS-Admin and 1234.

6. After logging in you will see a list view of the table that includes all of the columns in the table. We will clean this up as we customize our Xataface Moodle database backside in the next chapter. Be careful as you use the Xataface application to navigate through the table as the delete and update links found in the application will allow you to modify or delete information from the table. This data is used by Moodle and could be corrupted or lost permanently if you are not careful.

## *What just happened?*

We have just limited the tables that are displayed in our Xataface application by using the [_tables] directive and also prevented access to all tables that we don't want users to have access to using the [_disallowed_tables] directive. These settings, or directives, allow us to fine tune the display and access capabilities of our Xataface application.

## Time for action – another Xataface security measure

Before we finish this chapter, there is one additional security measure that should be introduced in order to ensure that your Xataface application is safe. We earlier created our admin account in the dataface__users table in the database and entered a simple password. We should, of course, use a more complex password and certainly recommend doing so on a live production website. It is also recommended that passwords be stored in an encrypted format in the database to prevent them from being viewed by anyone who has access to the database, legitimate or otherwise. Moodle encrypts passwords prior to inserting them into the database and Xataface can do so as well. Setup password encryption for your Xataface users table by following these steps:

1. Navigate to your Xataface application at yourmoodlesite/cims and log in using your admin account information: CIMS-Admin and 1234. Make sure you log in as admin before moving on to the next step. If you do not log in prior to completing the following steps, you will be unable to log in to your application.

2. In your CIMS directory, where you now have the following files and directories:
   - ❑ conf (directory)
   - ❑ conf.ini (file)
   - ❑ index.php (file)
   - ❑ Xataface (directory)

Create a directory titled `tables`.

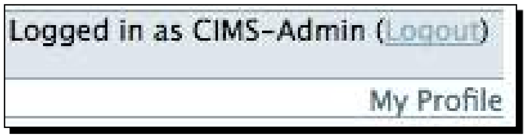
3. Inside the `tables` directory you just created, create a directory titled `dataface__users`. The directory name should be the same as the title you used for the table that houses the Xataface user information.

4. Now, inside the `dataface__users` directory create a file with your text editor and enter the following code:

```
[Password]
      encryption=md5
```

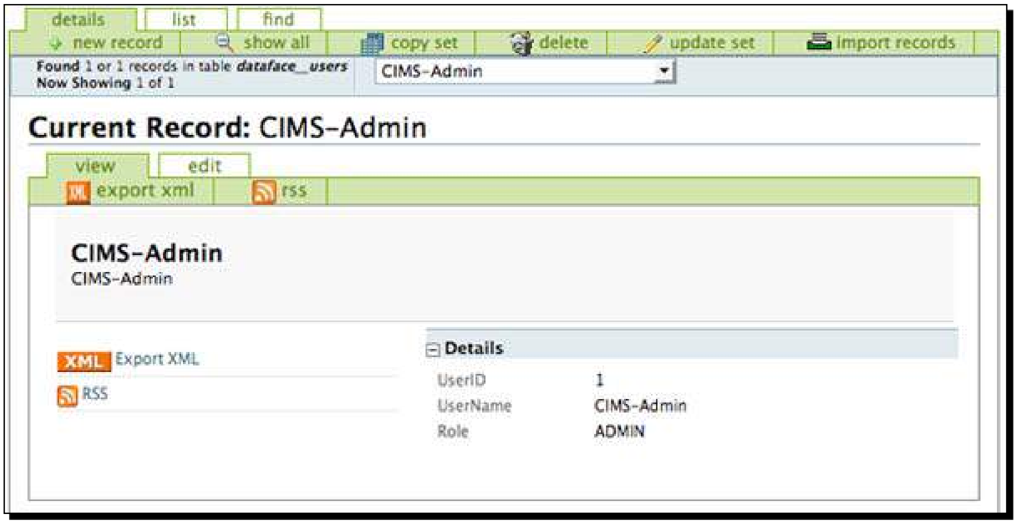
5. Save the file with the following file name: `fields.ini`.

6. In your Xataface application, that you logged into before completing the previous steps, click on the **My Profile** link found on the right side of the screen just below the **Logout** link, as shown in the following screenshot:



Logged in as CIMS-Admin (Logout)

7. Your profile, which consists of the information entered in the users table in the database will appear, as shown in the following screenshot. Click on the **edit** tab, enter a new password from the **edit** window and click on the **Save** button. (Note that clicking on the **new record** link here will allow you to create new user accounts.)



Found 1 of 1 records in table *dataface__users*
Now Showing 1 of 1

CIMS-Admin

Current Record: CIMS-Admin

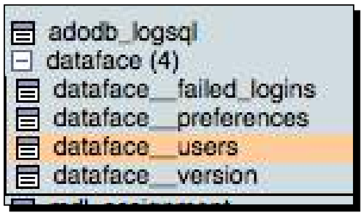
CIMS-Admin
CIMS-Admin

Details

| | |
|---|---|
| UserID | 1 |
| UserName | CIMS-Admin |
| Role | ADMIN |

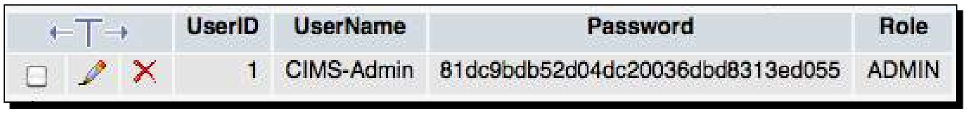
8. You can check to confirm that your password was encrypted by using PHPMyAdmin. Go to the front page of your Moodle site and click on the **Server** link and then on the **Database** link all found in the **Site Administration** block.

9. Click on the plus sign next to the **dataface** table from the table list on the left side of the page to expand the list of tables, as shown in the following screenshot:



adodb_logsql
dataface (4)
dataface__failed_logins
dataface__preferences
dataface__users
dataface__version

10. Click on the **dataface__users** table and then on the **Browse** link from the display window to the right of the table list. There should be only one entry in the table as we have only created the admin user. Notice that the password is now a long string of numbers and characters, as shown in the following screenshot. This confirms that Xataface is now encrypting the passwords stored in the user's table.



| ←T→ | UserID | UserName | Password | Role |
|---|---|---|---|---|
| □ | 1 | CIMS-Admin | 81dc9bdb52d04dc20036dbd8313ed055 | ADMIN |

## What just happened?

We have now performed some basic setup tasks for our Xataface application by activating the authentication plugin included in Xataface, specifying which tables, from the Moodle database, we want to be able to access via Xataface, and then by applying a setting to the **dataface__users** table that tells Xataface to encrypt all passwords inserted into the **Password** field in the users table. This very basic setup process only scratches the surface in terms of the customization Xataface is capable of. We will explore Xataface more in the next chapter.

## Pop quiz

1. How can new user accounts with encrypted passwords, for people who are to be given access to the Xataface application, be easily created?

   a. By adding the username and password information to the `conf.ini` file.

   b. By logging into the application as admin, clicking on profile and then on the new record link.

   c. By using PHPMyAdmin to insert a username, password, and role into the table.

   d. By clicking on the **Users**, **Accounts**, and then **Add a new user** link from inside the **Site Administration** block found on your Moodle front page.

# Summary

In this chapter, we installed four very useful tools that will help us to position our Moodle site solidly as a CIMS. The tools and the functions we have covered in this chapter are:

- **The Enhanced User Administration block**: This block allows you to sort and filter user accounts and perform various actions, bulk and single user, on the accounts. It can be a very handy tool for quickly retrieving simple user-related information from your Moodle site.
- **PHPMyAdmin**: We installed this very powerful database administration tool that is accessible via the Server link found in the Site Administration block of your site. We briefly demonstrated how this tool allows you to access and view all of the contents of the tables stored in the Moodle database.
- **Custom SQL Query tool**: This is a tool that is installed as a report and is accessible via the Reports link found in the Site Administration block on the front page of your Moodle site. We constructed one sample query that returns final grades from all courses on your site and thus demonstrated the usefulness of this tool to educational administrators.
- **Xataface**: We walked through the installation and a very basic setup process for this tool that allows for controlled and customizable backside access to the Moodle database.

We will continue to work with these tools, especially PHPMyAdmin and Xataface, in the next chapter, as we move into further enabling our Moodle site to function as a central portal for curriculum and information management.