# Exercise\_3\_1: Use phpMyAdmin with a database

In this exercise, you will use phpMyAdmin to review the databases that are used in your textbook, and test the types of SQL statements that you may need to use in your PHP applications.

## Instructions

1. Start with the exercise\_3\_1.zip file provided by your instructor. This zip file contains the create\_db.sql script provided by your textbook publisher. Unzip the exercise\_3\_1 file and put the .sql file on your desktop.
2. Start Apache and MySQL using the XAMPP control panel. Open up your browser and visit <http://localhost/phpmyadmin>
3. You will be automatically logged in as the user “root” on your local installation of XAMPP MySQL. This user is the super user / admin user for MySQL. Because you are “root” you will have full permission to every MySQL database on this machine. In the near future, you will log into the phpMyAdmin web application that is running on mis3501.temple.edu. There you will only have access to one database assigned to you.
4. Notice the buttons in the upper left hand corner of the screen. Home and Refresh will be especially helpful in the future.



1. Click on the Home button, then click on the Import tab. Use the Import feature to load and execute the create\_db.sql file. Hint, the “Go” button is located at the bottom of the page. You may need to scroll to see the “Go” button.
2. Notice the new schemas created on the left hand panel of phpMyAdmin. Notice that the phpMyAdmin application automatically collapses similarly named databases together to save space on the screen. This is presentation only. In the background my\_guitar\_shop1 and my\_guiar\_shop2 are two totally separate schemas.

|  |  |
| --- | --- |
|  |  |

1. Click on my\_guitar\_shop1, then click on the SQL tab.



1. Use the SQL command text area to enter statements that perform the following actions. Write your statements in the spaces provided, as indicated.
2. Select Statements
	1. Write a command that retrieves (hint: selects) all data from the product table

Write your SQL statement here:

|  |
| --- |
| select \* from products; |

* 1. Write a command that retrieves the **product name** of every item in the products table.

Write your SQL statement here:

|  |
| --- |
| select productName from products; |

* 1. Write a command that retrieves the **product name** of every item that has a list price greater than or equal to $1199.00.

Write your SQL statement here:

|  |
| --- |
| select productName from products where listPrice >= 1199; |

1. Selecting from Two Tables (Joins)
	1. Write a command that returns the product code of all products of the Category ‘Drums’. Assume that you do not know the categoryID of Drums.

Write your SQL statement here:

|  |
| --- |
| SELECT productCode FROM `products`,categories where products.categoryID = categories.categoryID and categoryName = 'Drums' |

Write the output of your statement here:

|  |
| --- |
| ludwigtama |

1. Insert and Delete
	1. Write a command that inserts a new product into the product table.

Write your SQL statement here:

|  |
| --- |
| INSERT INTO products(categoryID, productCode, productName, listPrice) VALUES (1, 'Air','Air Guitar', 1.50); |

* 1. Write a command that deletes the product that you just entered. Use the productID to identify the product.

Write your SQL statement here:

|  |
| --- |
| delete from products where productid = 11; |

1. Update Statements
	1. Write a command that updates the list price of product id 6 from $415.00 to $500.00.

Write the output of your statement here:

|  |
| --- |
| update products set listPrice = 500 where productid=6; |

1. When you are done, click on Home, my\_guitar\_shop1, the Operations tab, and “Drop the Database (DROP)”

2. Repeat this action for my\_guitar\_shop2.