Creating and altering database tables

This is an ungraded lab. Sections in green will be demonstrated by the instructor but are not required activities for students. All other sections should be completed either in pairs or individually.

Note: To get the documentation on any MySQL command, the easiest way is to use a web search for “mysql [command]”. For example, “mysql select” or “mysql group by”. Then click on the link at https://dev.mysql.com. Try this now. Another good way to understand a MySQL command is to use the graphical interface in phpMyAdmin to perform an operation. There is always some method of viewing the corresponding SQL, and this can give you useful examples for understanding the SQL. You already did this to understand the UPDATE command in assignment CC1, question B4. Below you will see other important examples, such as how to use ALTER TABLE.

# Part 1: Using MySQL from the command line

[Demonstration only; not required for students to do this.] Start MySQL within XAMPP without first starting Apache. Demonstrate that we cannot use phpMyAdmin. Demonstrate that we can still use MySQL from the command line (using cygwin terminal in this case:

/cygdrive/c/xampp/mysql/bin/mysql.exe -u root

Demonstrate creating and using a database:

SHOW DATABASES;

CREATE DATABASE carlisle\_menu\_items;

SHOW DATABASES;

Demonstrate basic SQL functionality:

USE wine;

SELECT \* FROM `product` WHERE `PRODTYPE`='red' AND `AVAILABLE\_QUANTITY`>120;

Show the distinct processes running MySQL Server and the MySQL client.

# Part 2: Creating and altering tables in phpMyAdmin

[Student activity starts here]

Create database temp\_menu using phpMyAdmin. Create a ‘restaurant’ table with appropriate columns for the restaurant:

Table ‘**restaurant’**

|  |  |  |
| --- | --- | --- |
| **id** | **name** | **address** |
| 23 | Issei Carlisle | 54 W High St, Carlisle, PA 17013 |
| 49 | Mt Fuji | 149 N Hanover St, Carlisle, PA 17013 |

What data types did you use for the columns? Experiment with changing them. So far, your table has just columns with no data in them.

[Instructor only] Use the SQL produced here (via Show SQL) to paste in to terminal for a different database e.g. temp\_menu\_items

Add a row of data to the restaurant table using phpMyAdmin: while browsing the table, choose the Insert tab. You can type in the data here. Ignore the Function column; type into the Value column.

View the corresponding SQL. If you need to go back and view it after the insertion, select the row, choose Copy, then Preview SQL.

Now delete the table completely and we will practice creating it from a file instead. To delete the table, browse the database and click on Drop. Note the corresponding SQL for this action.

Create a new Excel file, copy and paste all table data including the column names into the spreadsheet. Save as CSV. View the result in a text editor to make sure you understand the formatting. In phpMyAdmin, make sure you are browsing the carlisle\_menu\_items database then choose Import. Select the saved CSV file, check that all the options look sensible (the name of the table should be restaurant), then click Go. Take a look at the resulting table. Probably, the column names were imported as a tuple. We will delete that tuple soon.

Note that if we import the entire table without first creating the structure, we may need to make some adjustments to the data types. As an example here, change the restaurant ID to be an integer with up to seven digits. Note the SQL that was used:

ALTER TABLE `restaurant` CHANGE `id` `id` INT(7);

Note also that we will not be able to edit the content of the table until we have created a primary key. This is because phpMyAdmin needs a way to refer to specific tuples and there is no way to do that without a key. To add a primary key in phpMyAdmin, first browse the table then choose the Structure tab. Check the column or columns that will be the primary key, then click on Primary. Note the SQL that was used:

ALTER TABLE `restaurant` ADD PRIMARY KEY(`id`);

Delete the row that contains the column names. (Just click on Delete from the browse view.) Note the equivalent SQL:

DELETE FROM `restaurant` WHERE `restaurant`.`id` = 0;

Now suppose we wanted to create this table from the command line interface. We can find out the exact SQL needed to do that by using:

SHOW CREATE TABLE restaurant;

This should yield something like the following (some display options may need to be adjusted to view the entire result):

CREATE TABLE `restaurant` (

`id` int(7) NOT NULL,

`name` varchar(14) DEFAULT NULL,

`address` varchar(36) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8

[Instructor] Demonstrate how to do use this in the command line interface. To insert the data using the command line interface, the easiest thing will be to first export from phpMyAdmin. Browse to the restaurant table then click Export, and export to a .sql file. Open this file in a text editor, then we can copy and paste the INSERT command to insert the data from the command line. It should be something like the following:

INSERT INTO `restaurant` (`id`, `name`, `address`) VALUES

(23, 'Issei Carlisle', '54 W High St, Carlisle, PA 17013'),

(49, 'Mt Fuji', '149 N Hanover St, Carlisle, PA 17013');

[Students]

Now on your own, add the data from the menu items into a new table called item. Use the CSV file method; don’t type all the data in manually. Here is the data to be inserted:

Table ‘**item’**:

|  |  |  |
| --- | --- | --- |
| id | name | restaurant\_id |
| 6 | miso ramen | 23 |
| 18 | ebi tempura | 49 |
| 8 | tantan ramen | 23 |
| 18 | pad thai | 23 |
| 18V | vegetable pad thai | 23 |
| D4 | mabo tofu | 23 |
| 23 | temaki dinner | 49 |

Make appropriate changes to the data types, and add a primary key. (For the primary key, if you get an error message about a duplicate value, think about what column or columns you can select to obtain a unique value for every tuple.)

Now we need to link the two tables using a foreign key. Specifically, we want the restaurant\_id column in the item table to reference the id column in the restaurant table. To do this in phpMyAdmin, first browse the item table structure then click on Relation View. Add a new Foreign Key constraint by specifying a name for the constraint (e.g. restaurant\_id\_FK), then the column in the item table that will be a foreign key (restaurant\_id) then the name of the table and column which are referenced by our foreign key (restaurant and id). The ON DELETE and ON UPDATE options will be explained later. The equivalent SQL for adding the foreign key should be something like the following:

ALTER TABLE `item` ADD CONSTRAINT `restaurant\_id\_FK` FOREIGN KEY (`restaurant\_id`) REFERENCES `restaurant`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;

If there is an error about the foreign key constraint been violated, check the values of all of the restaurant\_id cells. Are they valid?

Now try to change a restaurant ID in the restaurant table. What happens?

The problem is that we specified ON UPDATE RESTRICT when we created the foreign key. This means we are restricted from updating values affected by the foreign key constraint. Change this to ON UPDATE CASCADE. Now try to change a restaurant ID in the restaurant table. It should be possible, and it should also cause corresponding changes in the item table.

Suppose we wanted restaurant names to be unique. We can enforce this in the Structure tab of the restaurant table. Check the name column and click Unique. Note the equivalent SQL. Now remove this uniqueness constraint by clicking Drop in the Indexes section of the Structure tab.

Finally, notice that we can provide default values for columns. As an example of this, change the name column in the item table to have the default value “unknown”. To do this, browse the table structure, and click on Change for the name column. Under default, select ‘As defined:’ then enter “unknown”. The equivalent SQL is something like:

ALTER TABLE `item` CHANGE `name` `name` VARCHAR(18) DEFAULT 'unknown';

To see the effect of this, browse the item table then click on the Insert tab to create a new item. You will see “unknown” has been entered as the default value.