

PROJECT 1

DATABASE SETUP (SEE CHAPTERS 1 AND 2) [0-10 POINTS]

In order to have data to use in a server-side database structure for this project 1 and project 2 (project 3 will use a different database), you will need to open the MySQL application that was installed with XAMPP on your PC or MAC and do the following:

1. Select the **Admin** button for MySQL from the control panel. A new window will open.
2. Select the **Databases** tab at the top left of this window.
3. Create a new database called *company* (with latin1_swedish_ci collation). Click Create. Make sure you are in this database. **Take a screenshot to verify and put into a document entitled *screenshots.docx* in the main folder.**
4. Next, click on the **SQL** tab at the top of the window (second tab).
5. In the first text box area at the top of this page, copy and paste the contents of the file called *Project1_Company_Creates_Inserts.docx* (just do a select all, then CTRL+C once in this file) into it.
6. Once all of the SQL code is in this text box area at the top of this page, click where it says *Go* in the lower right ...



7. Once this script runs, make sure all seven (7) tables in the database are present and that all inserts worked by clicking on the structure tab and browsing each table and its contents. Refer to Chapter 2 of the textbook for more information on how to maneuver in MySQL if needed or if issues arise. **Take a screenshot to verify and put into a document entitled *screenshots.docx* in the main folder.**

PROJECT LAYOUT [0-10 POINTS]

1. (0-5 points) You will model your files and site for this project after what you did in the previous assignment. Use the *index.php* includes file in the *sampleP1.zip* file to create a main page (**with a proper introduction [should be one to two paragraphs in length]**) where a navigation link exists for each of the queries and other specifics below. There is a sample query there as well. Look it over but delete it before submitting your work. **Show the exact question, the SQL query itself and the query results with proper labels for all output columns when the link is clicked. Name the link the same as the file name (without the .php extension). Put the links in the same order as the questions appear here.** Structure your code so that you use an **index.php** file, a **header.php** file, a **content.php** file, a **nav.html** file and a **footer.php** file as in the previous PHP includes template assignment. As before, there should be a main folder (called *<initials>p1* where *<initials>* is your initials ... i.e., mine would be *LLCp1*) with all of these files in

it. There should also be an *includes* folder that has six (6) php files in it that are named as specified below for the SQL Queries section (one for each of the queries below). **Show all outputs with correct formatting (i.e., currency, \$ where appropriate, etc.).**

- (0-5 points) Make sure you also have a database php file called *db.php* in the *includes* folder as well and set the password to an empty string "" in it (see zip file for sample). This php file should be included in the **index.php** file. Points will be deducted if this is not done! This is not negotiable.

SQL QUERIES (SEE CHAPTERS 2, 3 AND APPENDIX B) [0-60 POINTS]

A sample query has been completed for you. You can find it, along with the supporting *index.php* file, *db.php* file, etc. in the **sampleP1.zip** file under the Course Content -> Course Guides and Assignment Instructions -> Assignment Instructions -> Projects area at the course website to download (the name of the file is **sampleP1.zip**). The query in the sample zip file answers the question "What are the ids and names of every country in the countries table, ordered in ascending order by country id?". Use it as a starting point for the queries below. Be sure to remove it before submitting your work. See #1 in **Project Layout** above for details on what to include for each of these queries on the page/screen.

- (0-10 points) Write a query to display the names (last_name, first_name) from the **employees** table using an alias for the name "Last Name", "First Name". Order the results by last name in ascending order, then first name in ascending order (see Appendix B for *alias* and *order by* keywords). [call the file **DisplayNames.php**]
- (0-10 points) Write a query to get unique department IDs from the **employees** table using an alias name "Department ID". Order the results in descending order by Department ID. [call the file **EmployeeDepartment.php**]
- (0-10 points) Write a query to get all employee details from the **employees** table for employees whose last names start with an S. Order by last name, descending then by first name, descending. [call the file **EmployeeLastS.php**]
- (0-10 points) Write a query to get the names (last_name, first_name), salary, PF of all the employees (PF is calculated as 12% of salary) from the **employees** table. Order the results by last name in ascending order, then by first name in ascending order. [call the file **EmployeePF.php**]
- (0-10 points) Write a query to get the employee ID, names (first_name, last_name), salary from the **employees** table in ascending order of salary. Limit the output to only those employees with a salary greater than or equal to \$10,000. [call the file **EmployeeSalaries10000.php**]
- (0-10 points) Write a query to get the total salaries payable to employees from the **employees** table. This query should result in only one number being output. Label the output as *Total Salaries* and make sure the value is in currency format. [call the file **TotalSalaries.php**]

ADDING A PERSONALIZED WELCOME MESSAGE (SEE CHAPTER 3) [0-10 POINTS]

Add a personalized welcome message to the main pages of your web site (in the header portion of it) using an html form and php code that prompts the user for the *first name* and

last name and then outputs “Welcome to my site, Professor!” if the first name entered is your professor’s *first name* and the *last name* entered is your professor’s last name. Otherwise, output “Welcome to my site, *first name last name*!”. **The name should be permanent on all pages of the site in the header portion.** You will need to create a **name.html** file and a **name.php** file in the *includes* folder for this part of the project to add this code. You are also expected to do validation on the inputs.

ADDING AND DELETING A NEW USER (SEE CHAPTER 4) [0-10 POINTS]

1. (0-5 points) Add a new link after the last query called *Add Employee* on the main page that will open an html form to prompt the user for information needed to add a new user to the **employees** table. Take the information entered and add the new user to the **employees** table. Do a SQL select query on this newly added user (only) to verify it worked (using *employee_ID*) and output the results for this user only. **You should obtain the next available primary key value (i.e., incremented from the current highest value) via code and not allow the user to enter this in the form.** Do a SQL select query after the insert to verify it worked. Add the appropriate files to the *includes* folder for this part of the project. Be sure to ask for input for all attributes. Remember to validate your inputs on your form!
2. (0-5 points) Add a new link after #1 above called *Delete Employee* on the main page that will open an html form to prompt the user for information needed that will allow them to delete a user from the **employees** table. The delete should be done using the Primary Key of the user. Once the user enters this key, output **all** other field values in the **employees** table for that user and ask them to **verify** it is the correct user, then check response and either delete the user or don’t, based on their response. Do a SQL select query **before** and **after** the delete to verify it worked (last query should show no results). Give the user confirmation that it worked. Add the appropriate files to the *includes* folder for this part of the project.

WHAT TO SUBMIT

You will need to submit a zip file with all of these files included in it (including a Word document with screenshots as required). If you deviate from the above, then please include a *readme* text file in your folder, explaining your changes and why you made them (this should be minimal) and get approval from your instructor before submission. To test your code/site, I will be downloading this zip file and extracting all of your files and placing the **<initials>p1** folder (where *<initials>* is your initials ... i.e., mine would be **LLCp1**) in my XAMPP/htdocs folder on my PC to test them. You are encouraged to do the same so that you can guarantee your code will work in this environment. Errors and warnings will not be acceptable and will result in lost points. Make sure you have error checking turned on in your Php code! Failure to do this or to test your code to make sure it works will result in lost points for the assignment.

Keep in mind that all report outputs should be pleasing to the eye and presented to the user in such a way that all data is present and accurate (this includes labels for outputs and links to go from each page to other pages in your site, etc.).

P. S. All code should validate (HTML 5, CSS 3, Php) and should be efficient, working code (see <https://validator.w3.org/> and <http://jigsaw.w3.org/css-validator/>). For example, you

should not have CSS code that you do not use, your CSS code should be in an external file as specified above, all Php code should be structured well as is described throughout the chapters in the book, etc. Normal error checking should occur in your Php code and MySQL queries as well (as is demonstrated throughout the textbook). You won't be able to validate html code embedded into php files.

PROJECT 1 GRADING RUBRIC	
<u>SECTION</u>	<u>POSSIBLE POINTS</u>
DB SETUP	10
LAYOUT	
file structure, layout	5
root password	5
QUERIES	
display_names	10
employee_department	10
employee_last	10
employee_pf	10
total_salaries	10
unique_department	10
WELCOME MESSAGE	10
ADDING/DELETING USER	
add	5
delete	5
LATE?	
	100
attachments:	