

Database Things!

Remember: ANY time you touch a database, there should be a function created/modified in db.php.
NEVER put "SELECT *..." or ANY sql code in your web page file!

This lab will require the following activities/database changes and considerations:

- Updating a users password via **PREPARED STATEMENT**
- Having a URL to a client LOGO in the database. This will be stored as a **TEXT** field in the **CLIENTS** table that represents a LINK to an uploaded file in your directory. You will need to add this TEXT field to the table.
- Modify the code that CREATES a client in the database to accommodate the new Logo field.
Ex: INSERTS will need to be modified. You should **NOT** need to create a new prepared statement for this.
- You will need to be able to retrieve ALL Clients/SalesPeople(users)/Calls. You may have done some of this already.
- TO create PAGED results, there are a few ways you can do this. One of them is, you will **4 PHP FUNCTIONS** for retrieving **PAGED** record results Clients/users/salespeople/calls. You can instruct PSQL to retrieve a subset (like a page isn't it?) of a full result set using LIMIT and OFFSET.

```
pg_prepare($dbconn, "client_select_all", 'SELECT id, email, first_name, last_name, phone_number, extension, logo_path FROM clients LIMIT $1 OFFSET $2');
```

- The LIMIT \$1 (\$1 becomes a value when you pass in a parameter on pg_Execute) represents HOW MANY to take
- The OFFSET \$2 (\$2 becomes a value when you pass in a parameter on pg_Execute) represents essentially WHERE TO START in the full result.

Consider the following example:

```
SELECT * from [Users] where type='a'
```

And assume this returns 10 results.

- if I say LIMIT 5 OFFSET 0, it would ONLY return the FIRST 5.
- if I say LIMIT 5 OFFSET 5 it will return ONLY the LAST 5.

- Alternatively, you can return the total result set (not use OFFSET or LIMIT) and then "post process" the result in php code using [ARRAY FUNCTIONS](#) where you might take 10 at offset 10
- So a function call for this *might* look like this:

```
client_select_all($page),
```

Update Password Page




Now that users (salespeople) can be created, we must give them the ability to change their own password! This will be done via a **NEW PAGE** named updatepassword.php. This page will only be accessible by ANY TYPE of LOGGED IN user.

YOU ARE TO:

- You MUST use the display_form function you created in lab2 to display two input fields
 - New Password
 - Confirm Password
 - And button if you were putting buttons in the function call.
- The page will use a SELF referring form (don't post to any other page)
- There will be the following validation rules:
 - Both fields are REQUIRED
 - Both fields must match eachother
 - **FORM MUST NOT BE STICKY**
- If validation errors exist:
 - Any or all Error messages must flash (show up ONCE.) on the page
 - Password will NOT be updated
- If no validation errors exists:
 - A success message must flash (show up ONCE.)
 - User entered password will be ENCRYPTED
 - Encrypted password will be used to UPDATE the user's password field
- NO OTHER FIELDS ARE TO BE UPDATED OR OTHERWISE CHANGED HERE
- Navigation and access will be as follows:
 - This will only be accessible when logged in by ANY user type, otherwise redirect to sign-in.php with message.
 - Menu item only shows when logged in, all user types.

Client Logo

You already have the ability to create Clients. You will be MODIFYING that functionality to allow for adding an image file of their logo. This will require a file input in your form. Consider placing it here.

First Name	
Last Name	
Email	
Phone	
Extention	
Salesperson	
	
	

Information on the file input can be found in the File_upload powerpoint lecture from week09 or on [w3schools](https://www.w3schools.com).

As per the lecture, you will want to be sure you are doing some form of validation on the file and saving it to a proper folder on the server. **I Recommend** using your_lab_folder/images. This step would be done using code similar to:

```
if (isset($file)) {  
    $tmp_name = $file["tmp_name"];  
    $name = basename($file["name"]);  
    move_uploaded_file($tmp_name, "./upload/$name");  
}
```

Displaying Data Tables

When we want to display records from database in a USER FACING INTERFACE, we don't want to display them as a messy, squished, unformatted blob of text.

It is better to format and list the information in a structured, formatted table like this:

Name	Position	Office	Age	Start Date	Salary
Airi Satou	Accountant	Tokyo	33	2008-11-28	\$162,700.00
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009-10-09	\$1,200,000.00
Ashton Cox	Junior Technical Author	San Francisco	66	2009-01-12	\$86,000.00
Bradley Greer	Software Engineer	London	41	2012-10-13	\$132,000.00

source: <https://datatables.net/extensions/searchbuilder/examples/initialisation/html.html>

PLEASE NOTE: I do not expect you to add any of the functionality this table has. You do not need to SORT or colour code anything.

When display database records where there could be **MANY** records, we do not want to return ALL of them ALL of the time. If we had 150,000 calls in the calls table, we wouldn't want to load a table of 150,000 calls to the browser.

Modern websites use PAGINATION for their tables. This means having the table only display a predefined number of records from within a full results set. If we have 150,000 calls, we might show them 50 **at a time** and have the ability to move through the pages of data.

[MORE INFO ON PAGINATION](#)

YOU ARE TO:

- For each of the pages where we create data (salespeople, clients, calls) you MUST display the records in the DATABASE table, in a nicely structured, formatted HTML TABLE.
- This will utilize a NEW function in functions.php called:
display_table(\$fieldNames, \$records, \$totalRecords, \$currentPage);
 - **\$fieldNames:** is an array[] of strings that represent the database field name AND the human readable label that will go in the table header
 - **\$records:** is an array of Clients/salespeople/calls/whatever we end up doing that will be each put into a table row <tr><td></td></tr>
 - **\$totalRecords:** is the total number of records your query would return 100 records
 - **\$currentPage:** this is the page in the db result set you are on and NOT the webpage you are on.
- This will use your code for retrieving clients as mentioned in the DB section. You might use SQL abilities such as OFFSET and LIMIT, OR you would return the whole result set and use [ARRAY FUNCTIONS](#)
- For the clients.php datatable, you will have to remember to properly size the image so that a MASSIVE image will not break your table. You could use something like [THIS on w3schools](#)