

CMPS 3680 Lab 6

Spring 2026

Part 1 - Setup

1. For this assignment you will start with your completed lab5, so I would recommend making a copy of it.
2. To do this, run the following command from inside your cs3680 folder:

```
cp -r lab5/ lab6/  
cd lab6/
```

Part 2 - sqlTools.php (continued)

1. You should already have a working `getConnection` and `closeConnection` in this file from Lab 5.
2. Define the function `addProduct($product)` to do the following:
 - Get a database connection from `getConnection`
 - Use `mysqli_prepare`, `mysqli_stmt_bind_param`, `mysqli_stmt_execute` to add the data passed `$product` to the database. Remember that the original `product['id']` from the html form should map to `internalId` in the database.
 - Be sure to check for errors along the way and use `die()` to output any error messages
 - Close the database connection with `closeConnection`
3. Define the function `removeProduct($id)` to do the following:
 - Get a database connection from `getConnection`
 - Use `mysqli_prepare`, `mysqli_stmt_bind_param`, `mysqli_stmt_execute` to remove the data passed `$product` to the database. Remember that the original `product['id']` from the html form should map to `internalId` in the database.
 - Be sure to check for errors along the way and use `die()` to output any error messages
 - Close the database connection with `closeConnection`
4. Define the function `dumpProducts()` to do the following:
 - Get a database connection from `getConnection`
 - Call `mysqli_query` to execute the query string `'DELETE FROM product;'`
 - Be sure to check for errors along the way and use `die()` to output any error messages
 - Close the database connection with `closeConnection`
5. Define the function `getProducts()` to do the following:
 - Get a database connection from `getConnection`
 - Use `mysqli_prepare`, `mysqli_stmt_bind_param`, `mysqli_stmt_execute` to retrieve all of the product records from the database.
 - Loop through the rows returned by the query and build an array that matches the output returned by the original `getProducts` function in `productTools.php`
 - Close the database connection with `closeConnection`
6. Define the function `uniqueID($id)` to do the following:
 - Get a database connection from `getConnection`
 - Use `mysqli_prepare`, `mysqli_stmt_bind_param`, `mysqli_stmt_execute` to see if any product in the database has an `internalId` matching `$id`.
Hint: `mysqli_stmt_store_result` and `mysqli_num_rows` may come in handy.
 - Return true if 0 records/rows are found, and false otherwise.
 - Close the database connection with `closeConnection`

Part 3 - The Swap

1. Once you have completed and tested all of the functions in Part 2, you should be able to start using it in your original `add.php` and `remove.php` files.
2. To do this, simply replace `require_once "productTools.php"` with `require_once "sql/sqlTools.php"`
3. As long as you used the same function names in your `sqlTools` file, and you made sure to map the original `$id` from the form to `internalId` in your table, everything should just work.