

Programmer's Guide for "A Database Approach to World War I"

There are six main pages:

index.php
dynamixIndex.php
createYourOwn.php
result.php
queries.php
templates.php

The index, dynamic index, and create your own pages are the user front end for selecting the query type or giving user input. The result page is what is loaded after the user submits a form on any of the above three pages. All three index pages go to the same result page. The templates pages stores the header that is the same for every page and for any additional page you create it should require templates.php and call `getHeader()`. To add a static query all you need to do is add to the static array stored in the queries.php and a radio button will be added to the index page and the results will generate themselves without additional code.

Note: the table headers created for the results come from the SQL so in the SELECT clause if you would like to give it a nice heading title you have to rename the attributes using 'as' if you would like spaces in the header you can say things like: num_casualties as 'Number of Casualties' and it will format appropriately.

To add a dynamic query you have to add the SQL and title to display to the other array in queries.php for dynamic queries. The user input must be sanitized and the name of the input that holds the user input must be put into the array. As such you have to manually add the query to the dynamicIndex.php this includes adding the radio button with the appropriate name (the one that matches what you put in the queries.php) and an input that matches the appropriate name (the one you put into queries.php).

Results are generated dynamically and the same way for every type of query that way the result generation code does not have to change for each new query. First a database connection is made and the appropriate array of queries is loaded from queries.php. At this point the link that is put in after the results is created to go back to the page from which the user came ie if they were on the dynamic index and go to results the Choose Another query will go back to the dynamic index instead of say just index every time.

After the queries are loaded the appropriate query is run and the result fields are looped through to generate the table header row. Then the resulting rows are looped through to generate each table row. After the query logic has occurred the html that was generated is echoed out and the database connection is closed.

If an error occurred it should generate the header and the back button but instead of a table it should display the error.

Method summary:

Page: templates.php

getHeader()

- takes no input and returns an html screen to echo out the header of the page
- this method is called by every page including results

getDropdown(\$name)

- takes a name for the html element generates a dropdown from the list of participants in the database with that name. The name is important because the value of the dropdown at the time of the form submission can be accessed through the name given to the input ie \$_POST[\$name] will contain the value of the dropdown when the user submits a form.
- this method is currently only used by the dynamic index page

Page: queries.php

getStaticQueries()

- takes no input and returns an array of queries that are static ie do not require user input to display results
- Each entry of the array is an array containing the title and the SQL to run the query. For example to list all battles the "query type" is 'listBattles' so the array queries has an entry called 'listBattles' that is:

```
$queries['listBattles'] = array();
```

```
$queries['listBattles']['title'] = "List of Battles";
```

```
$queries['listBattles']['sql'] = "SELECT b.start_date, b.end_date, b.title, b.location FROM battle as b";
```

- this array currently generates the radio buttons on the index page and the list of possible SQL queries for the List all Queries query.

getDynamicQueries(\$db)

- takes a database connection (for use in sanitizing the user input string)
- returns an array similar to the static queries
- The dynamic part comes from an additional field in the array that can be used to place the sanitized user input into the SQL. For example:

```
$queries = array();
```

```
$queries['byLocation'] = array();
```

```
$queries['byLocation']['userInput'] = mysqli_real_escape_string($db, $_POST['location']);
```

```
$queries['byLocation']['title'] = "List of Battles in Location: " . $queries['byLocation']['userInput'];
```

```
$queries['byLocation']['sql'] = "SELECT b.start_date, b.end_date, b.title, b.location FROM battle as b  
WHERE b.location LIKE '%". $queries['byLocation']['userInput'] . "%";
```