**Message Board**

**Example chapter**

Making the database

* The languages table stores the name of a language in its own alphabet and in English
* The threads table relates to the languages and users table (each message can only be in one forum and by one user; each forum can have multiple messages, and each user can post multiple messages).
* The posts table stores the actual bodies of the messages associated with a thread.
* The language, thread and post tables provide the bulk of the forum functionality
* The users table only allows registered users to post messages. A combination of their username and password would be used to login.
* The words table is necessary to make the site multilingual. This table will store translations of common elements: navigation links, form prompts, headers etc…

In MySQL (PHPMYADMIN):

1. Set the character set to utf8
2. CREATE DATABASE forum CHARACTER SET UTF8;

USE FORUM

1. Create the languages table

CREATE TABLE languages(

Lang\_id TINYINT UNSIGNED NOT NULL AUTO\_INCREMENT,

Lang VARCHAR(60) NOT NULL,

Lang\_eng VARCHAR(20) NOT NULL,

PRIMARY KEY (lang\_id),

UNIQUE (lang)

);

//the language\_eng column is the English equivalent of the language so that the administrator can easily see which language is which.

1. Create the threads table

CREATE TABLE threads (

Thread\_id INT UNSIGNED NOT NULL AUTO\_INCREMENT,

Lang\_id TINYINT(3) UNSIGNED NOT NULL,

User\_id INT UNSIGNED NOT NULL,

Subject VARCHAR(150) NOT NULL,

PRIMARY KEY (thread\_id),

INDEX (lang\_id),

INDEX (user\_id)

);

//the columns that will be used in joins and WHERE clauses -lang\_id and user\_id – are indexed, as is thread\_id (as primary key, it will be indexed)

1. Create the posts table

CREATE TABLE posts (

Post\_id INT UNSIGNED NOT NULL AUTO\_INCREMENT,

Thread\_id INT UNSIGNED NOT NULL,

User\_id INT UNSIGNED NOT NULL,

Message TEXT NOT NULL,

Posted\_on DATETIME NOT NULL,

PRIMARY KEY (post\_id),

INDEX (thread\_id),

INDEX user\_id)

);

1. Create the users table

CREATE TABLE users (

User\_id MEDIUMINT UNSIGNED NOT NULL AUTO\_INCREMENT,

Lang\_id TINYINT UNSIGNED NOT NULL,

Time\_zone VARCHAR(30) NOT NULL,

Username VARCHAR(30) NOT NULL,

Pass CHAR(40) NOT NULL,

Email VARCHAR(60) NOT NULL,

PRIMARY KEY (user\_id),

UNIQUE (username),

UNIQUE (email),

INDEX login (username, pass)

);

//you could also omit some other columns such as first name, surname, date registered etc…

1. Create the words table

CREATE TABLE words (

Word\_id TINYINT UNSIGNED NOT NULL AUTO\_INCREMENT,

Lang\_id TINYINT UNSIGNED NOT NULL,

Title VARCHAR(80) NOT NULL,

Intro TINYTEXT NOT NULL,

Home VARCHAR(30) NOT NULL,

Forum\_home VARCHAR(40) NOT NULL,

`language` VARCHAR(40) NOT NULL,

Register VARCHAR(30) NOT NULL,

Login VARCHAR(30) NOT NULL,

Logout VARCHAR(30) NOT NULL,

New\_thread VARCHAR(40) NOT NULL,

Subject VARCHAR(30) NOT NULL,

Posted\_on VARCHAR(30) NOT NULL,

Posted\_by VARCHAR(30) NOT NULL,

Replies VARCHAR(30) NOT NULL,

Latest\_reply VARCHAR(40) NOT NULL,

Post\_a\_reply VARCHAR(40) NOT NULL,

PRIMARY KEY (word\_id),

UNIQUE (lang\_id)

);

//language is surrounded by backticks so that MySQL doesn’t confuse the columns name with the keyword *language*.

//\*body and submit missing from columns! \*

1. Populate the languages table

INSERT INTO languages (lang, lang\_eng) VALUES

(“English”, “English”),

(“Portugues”, “Portuguese”),

(“Francais”, “French”),

(“Norsk”, “Norwegian”),

(“Nederlands”, “Dutch”),

(“Deutsch”, “German”);

1. Populate the users table

INSERT INTO users (lang\_id, time\_zone, username, pass, email) VALUES

(1, “America/New\_York”, “trouster”, SHA1(“password”), email@example.com),

(6, “Europe/Berlin”, “Ute”, SHA1(“pa24word”), “email1@example.com”);

1. Populate the words table

INSERT INTO words VALUES

(NULL, 1, “PHP and MySQL for dynamic web sites: The forum!”, “<p>Welcome to our site… please use the links above… </p>”, “home”, “forum home”, “language”, “register”, “login”, “logout”, “new thread”, “Subject”, “body”, “submit”, “posted on”, “posted by”, “Replies”, “Latest reply”, “Post a reply”);

Writing the templates

This example, like any site containing lots of pages, will make use of a template to separate out the bulk of the presentation from the logic.

To create the templates (header and footer files), do the following:

Header:

1. Create new document called ‘header.html’
2. Start a session
3. Include the database connection
4. Determine the language ID
5. Get the keywords for this language
6. If the query returned no records, get the default words
7. Fetch the retrieved words into an array, free the resources, and close the PHP section
8. Start the HTML page and add CSS
9. Complete the HTML head and begin the body

Footer:

1. Create new document called ‘footer.html’
2. Complete the HTML page body
3. Place header.html and footer.html in web directory

Header:

<?php

Header(‘Content-type: text/HTML; charset=”UTF-8”);

//2.

Session\_start();

//for testing purposes

$\_SESSION[‘user\_id’] = 1;

$\_SESSION[‘user\_tz’] = “America/New\_York”;

//for logging out

$\_SESSION = array();

//3.

Require(‘../mysqli\_connect.php’);

//4.

If( isset($\_GET[‘lid]) && filter\_var($\_GET[‘lid’], FILTER\_VALIDATE\_INT, array(‘min\_range’ => 1)) ) {

$\_SESSION[‘lid’] = $\_GET[‘lid’];

} elseif (!isset($\_SESSION[‘lid’])) {

$\_SESSION[‘lid’] = 1; //default

}

//5.

$q = “SELECT \* FROM words WHERE lang\_id = {$\_SESSION[‘lid’]}”;

$r = mysqli\_query($dbc, $q);

//6.

If(mysqli\_num\_rows($r) == 0) { //invalid message id

//use the default language

$\_SESSION[‘lid’] = 1; //default

$q = “SELECT \* FROM words WHERE lang\_id = {$\_SESSION[‘lid’]}”;

$r = mysqli\_query($dbc, $q);

}

//7.

$words = mysqli\_fetch\_array($r, MYSQLI\_ASSOC);

Mysqli\_free\_result($r); //free the results

?>

//8. & 9.

<!DOCTYPE html>

<html lang=”en”>

<head>

<meta charset=”utf-8” />

<title> <?php echo $words[‘title’]; ?> </title>

//add js and style links here if needed

</head>

<body>

<table>

<tr>

<td> <?php $words[‘title’]; ?> </td>

</tr>

<tr>

<td> <?php //display links

Echo “<a href=’index.php’>” .$words[‘home’] .”</a>

<br /><a href=’forum.php’>” .$words[‘forum\_home’] “</a><br />”;

//display links based upon login status:

If(isset($\_SESSION[‘user\_id’])) {

//if this is the forum page, add a link for posting new threads:

If(basename($\_SERVER[‘PHP\_SELF’]) == ‘forum.php’) {

Echo “<a href=’post.php’>” .$words[‘new\_thread’] . “<a/><br />”;

}

//add the logout link:

Echo “<a href=’logout.php’>” .$words[‘logout’] . “</a><br />”;

}else{

//register and login links:

Echo “<a href=’register.php’>” .$words[‘register’] . “</a><br /> <a href=’login.php’> “ .$words[‘login’] . “</a><br />”;

}

//for choosing a forum/language:

Echo “<p><form action=’forum.php’ method=’get’>

<select name=’lid’>

<option value=’0’>” .$words[‘language’] .”</option>”;

//retrieve all the languages

$q = “SELECT lang\_id, lang FROM languages ORDER BY lang\_eng ASC”;

$r = mysqli\_query($dbc, $q);

If(mysqli\_num\_rows($r) > 0) {

While($menu\_row = mysqli\_fetch\_array($r, MYSQLI\_NUM)) {

Echo “<option value=\”$menu\_row[0]\”>$menu\_row[1]</option>\n”;

}

}

Mysqli\_free\_result($r);

Echo “</select><br />

<input name=’submit’ type=’submit’ value=’” .$words[‘submit’] . “’ /></form></p>

</td>

<td>”;

?>

Footer:

</td>

</tr>

<tr>

<td>&copy; 2018 Ali Issaee</td>

</tr>

</table>

</body>

</html>

Creating the index page (index.php)

The index page in this example won’t do much. It will provide some introductory text and the links for the user to register, log in, choose the preferred language/forum, and so forth.

<?php

Include(‘includes/header.html’);

Echo $words[‘intro’]; //the words array is defined within the header file

Include(‘includes/footer.html’);

?>

Creating the forum page (forum.php)

This page displays the threads in a forum (each language being its own forum). The page will use the language ID, passed to this page in a URL and/or stored in a session, to know what threads to display.

\*The query this page uses is perhaps the most complex one yet\*

<?php

Include(‘includes/header.html’);

//if the user is logged in and has chosen a time zone, use that to convert the dates and times:

If(isset($\_SESSION[‘user\_tz’])) {

$first = “CONVERT\_TZ(p.posted\_on, ‘UTC’, ‘{$\_SESSION[‘user\_tz’]}’)”;

$last = “CONVERT\_TZ(p.posted\_on, ‘UTC’, ‘{$\_SESSION[‘user\_tz’]}’)”;

}else{

$first = “p.posted\_on”;

$last = “p.posted\_on”;

}

$q = “SELECT t.thread\_id, t.subject, username, COUNT(post\_id) -1 AS responses, Max(DATE\_FORMAT($last, ‘%e-%b-%y %l:%i %p’)) AS last, MIN(DATE\_FORMAT($first, ‘%e-%b-%y %l:%i %p’)) AS first FROM threads AS t INNER JOIN posts AS p USING (thread\_id) INNER JOIN users AS u ON t.user\_id = u.user\_id WHERE t.lang\_id = {$\_SESSION[‘lid’]} GROUP BY (p.thread\_id) ORDER BY last DESC”;

$r = mysqli\_query($dbc, $q);

If(mysqli\_num\_rows($r) > 0) {

//create table

Echo ‘<table>

<tr>

<td>’ .$words[‘subject’] . ‘</td>

<td>’ .$words[‘posted\_by’] . ‘</td>

<td>’ .$words[‘posted\_on’] . ‘</td>

<td>’ .$words[‘replies’] . ‘</td>

<td>’ .$words[‘latest\_reply’] . ‘</td>

</tr>’;

//fetch each thread

While($row = mysqli\_fetch\_array($r, MYSQLI\_ASSOC)) {

Echo ‘<tr>

<td>’ .$row[‘username’] . ‘</td>

<td>’ .$row[‘first’] . ‘</td>

<td>’ .$row[‘responses’] . ‘</td>

<td>’ .$row[‘last’] . ‘</td>

</tr>’;

}

Echo ‘</table>’;

}else{

Echo ‘<p>There are currently no messages in this forum</p>’;

}

Include(‘includes/footer.html’);

The query in the example above needs to return 6 things: the ID and subject of each thread (which comes from the threads table), the name of the user who posted the thread in the first place (from users), the number of replies to each thread, the date the thread was started, and the date the thread last had a reply (all from posts).

The overarching structure of this query is a join between threads and posts using the thread\_id column (which is the same in both tables). The result is then joined with the users table using the user\_id column.

As for the selected values, three aggregate functions are used: COUNT(), MIN() and MAX(). Each is applied to a column in the posts table, so the query has a GROUP BY(p.thread\_id) clause. MIN() and MAX() are used to return the earliest and latest dates. The latest date is also used to order the results so that the most recent activity always gets returns first. The COUNT() function is used to count the number of posts in a given thread. Because the original post is also in the posts table, it will be factored into COUNT() as well, so 1 is subtracted from the value.

Creating the threads page (read.php)

This page is for viewing all of the messages in a thread. It is accessed by clicking a link on the forum page.

<?php

Include(‘includes/header.html’);

//check for a thread ID

$tid = FALSE; //flag variable – false as a way of saying ‘prove that the thread ID is valid’

If(isset($\_GET[‘tid’]) && filter\_var($\_GET[‘tid’], FILTER\_VALIDATE\_INT, array(‘min\_range’ => 1)) ) {

//create a shorthand version of the thread ID:

$tid = $\_GET[‘tid’];

//convert the date if the user is logged in:

If(isset($\_SESSION[‘user\_tz’])) {

$posted = “CONVERT\_TZ(p.posted\_on, ‘UTC’, ‘{$\_SESSION[‘user\_tz’]}’ )”;

}else{

$posted = ‘p.posted\_on’;

}

//run the query

$q = “SELECT t.subject, p.message, username, DATE\_FORMAT($posted, ‘$e-$b-$y %l:%i %p’) AS posted FROM threads AS t LEFT JOIN posts AS p USING (thread\_id) INNER JOIN users AS u ON p.user\_id = u.user\_id WHERE t.thread\_id = $tid ORDER BY p.posted\_on ASC”;

$r = mysqli\_query($dbc, $q);

If(!(mysqli\_num\_rows($r) > 0)) {

$tid = FALSE; //invalid thread ID!

}

} //end of isset ($\_GET[‘tid’]) IF.

If($tid) { //get the messages in this thread

$printed = FALSE; //Flag variable

//fetch each:

While($messages = mysqli\_fetch\_array($r, MYSQLI\_ASSOC)) {

//only need to print the subject once!

If(!$printed) {

Echo “<h2>{$messages[‘subject’]}</h2>\n”;

$printed = TRUE; //set to true so subject is not printed again.

}

//print the message

Echo “<p>{$messages[‘username’]} ({$messages[‘posted’]}) <br /> {$messages[‘message’]}</p><br />\n”;

} //end of while loop

//show the form to post a message:

Include(‘includes/post\_form.php’);

}else{ //invalid thread ID

Echo “<p>This page has been accessed in error</p>”;

}

Include(‘includes/footer.html’);

?>

The query above is like the query on the forum page, but its been simplified in two ways. First, it doesn’t use any of the aggregate functions or a GROUP BY clause. Second, it only returns one date and time. The query is still a JOIN across three tables, in order to get the subject, message bodies, and usernames. The records are ordered by their posted dates in ascending order (i.e., from the first post to the most recent).

If the query doesn’t return any rows, then the thread ID isn’t valid and the flag variable is made FALSE again.

Posting messages

2 files are needed to post messages:

1. One to make the form
2. One to handle the form

*Creating the form (post\_form.php):*

1. It can only be included by other files, never accessed directly
2. It should only be displayed if the user is logged in
3. If its being used to add a reply to an existing message, it only needs a message body input
4. If its being used to create a new thread, it needs both subject and body inputs
5. It needs to be sticky

<?php

//redirect if this page is called directly

If(!isset($words)) {

Header(‘location: http://www.example.com/index.php’);

Exit();

}

//only display this form if the user is logged in

If(isset($\_SESSION[‘user\_id’])) {

//display the form

Echo ‘<form action=”post.php” method=”post” accept-charset=”utf-8”>’;

//if on read.php

If(isset($tid) && $tid) {

//print a caption

Echo ‘<h3>’ .$words[‘post\_a\_reply’] .’</h3>’;

//add the thread ID as a hidden input:

Echo ‘<input name=”tid” type=”hidden” value=”’ .$tid .’” />’;

}else{ // new thread

//print a caption:

Echo ‘<h3>’ .$words[‘new\_thread’] .’</h3>’;

//create subject input

Echo ‘<p>’ .$words[‘subject’] . ’: <input name=”subject” type=”text” ‘;

//check for existing value

If(isset($subject)) {

Echo “value=\”$subject\” “;

}

Echo ‘/></p>’;

} //end of $tid IF.

//create the body text area:

Echo ‘<p>’ .$words[‘body’] . ‘: <textarea name=”body”>’;

If(isset($body)) {

Echo $body;

}

Echo ‘</textarea></p>’;

//finish the form:

Echo ‘<input name=”submit” type=”submit” value=”’ .$words[‘submit’] . ‘” />

</form>’;

}else{

Echo “<p>You must be logged in to post messages. </p>”;

}

?>

*Handling the form (post.php)*:

This page will be called in 3 different ways:

1. To handle the form for a thread reply (POST)
2. To display the form for a new thread submission (GET)
3. To handle the form for a new thread submission (POST)

If a new thread is being created, two queries must be run: one to add the thread to the threads table, and a second to add the new thread body to the posts table. If the submission is a reply to an existing thread, then only one query is required, inserting a record into posts.

<?php

Include(‘includes/header.html’);

If($\_SERVER[‘REQUEST\_METHOD’] == “POST”) {

//language ID is in the session.

//validate thread ID ($tid), which may not be present:

If(isset($\_POST[‘tid’]) && filter\_var ($\_POST[‘tid’], FILTER\_VALIDATE\_INT, array(‘min\_range’ => 1)) ) {

$tid = $\_POST[‘tid’];

}else{

$tid = FALSE;

}

//if there’s no thread ID, a subject must be provided

If(!$tid && empty($\_POST[‘subject’])) {

$subject = FALSE;

Echo ‘<p>Please enter a subject for this post.</p>’;

}elseif(!$tid && !empty($\_POST[‘subject’])) {

$subject = htmlspecialchars(strip\_tags($\_POST[‘subject’]));

}else{ //thread ID, no need for subject

$subject = TRUE;

}

//validate the body:

If(!empty($\_POST[‘body’])) {

$body = htmlspecialchars(strip\_tags($\_POST[‘body’]);

}else{

$body = FALSE;

Echo “<p>Please enter a body for this post.</p>”;

}

If($subject && $body) { //ok

//add the message to the database

If(!$tid) { //create a new thread id

$q = “INSERT INTO threads (lang\_id, user\_id, subject) VALUES ({$\_SESSION[‘lid’]}, {$\_SESSION[‘user\_id’]}, ‘” . mysqli\_real\_escape\_string($dbc, $subject) .”’)”;

$r = mysqli\_query($dbc, $q);

If(mysqli\_affected\_rows($dbc) == 1) {

$tid = mysqli\_insert\_id($dbc);

}else{

Echo “<p>Your post could not be handled due to a system error </p>”;

}

}//no $tid

If($tid) { //add this to the replies table:

$q = “INSERT INTO posts (thread\_id, user\_id, message, posted\_on) VALUES ($tid, {$\_SESSION[‘user\_id’]}, ‘” . mysqli\_real\_escape\_string($dbc, $body) . “’, UTC\_TIMESTAMP())”;

$r = mysqli\_query($dbc, $q);

If(mysqli\_affected\_rows($dbc) == 1) {

Echo “<p>Your post has been entered</p>”;

}else{

Echo “<p>Your post could not be handled due to a system error. </p>”;

}

}

}else{ //include the form

Include(‘includes/post\_form.php’);

}

}else{ //display the form:

Include(‘includes/post\_form.php’);

}

Include(‘includes/footer.html’);

?>