# HACK-STACK Setup Part 3

## Set up a sessions database for our stack.

We want to set up sessions handler database store using our mySQL server for our PHP sessions data. The philosophy of why this might be of benefit is beyond our discussion, but there are various strong reasons that you might want to do this.

This requires a sessions handler function to interface the PHP with the sessions database. We use the **Zebra_Session** code which is described as:
"A drop-in replacement for PHP's default session handler which stores session data in a MySQL database, providing both better performance and better security and protection against session fixation and session hijacking"
The code can be found on the Zebra_Session author's page.

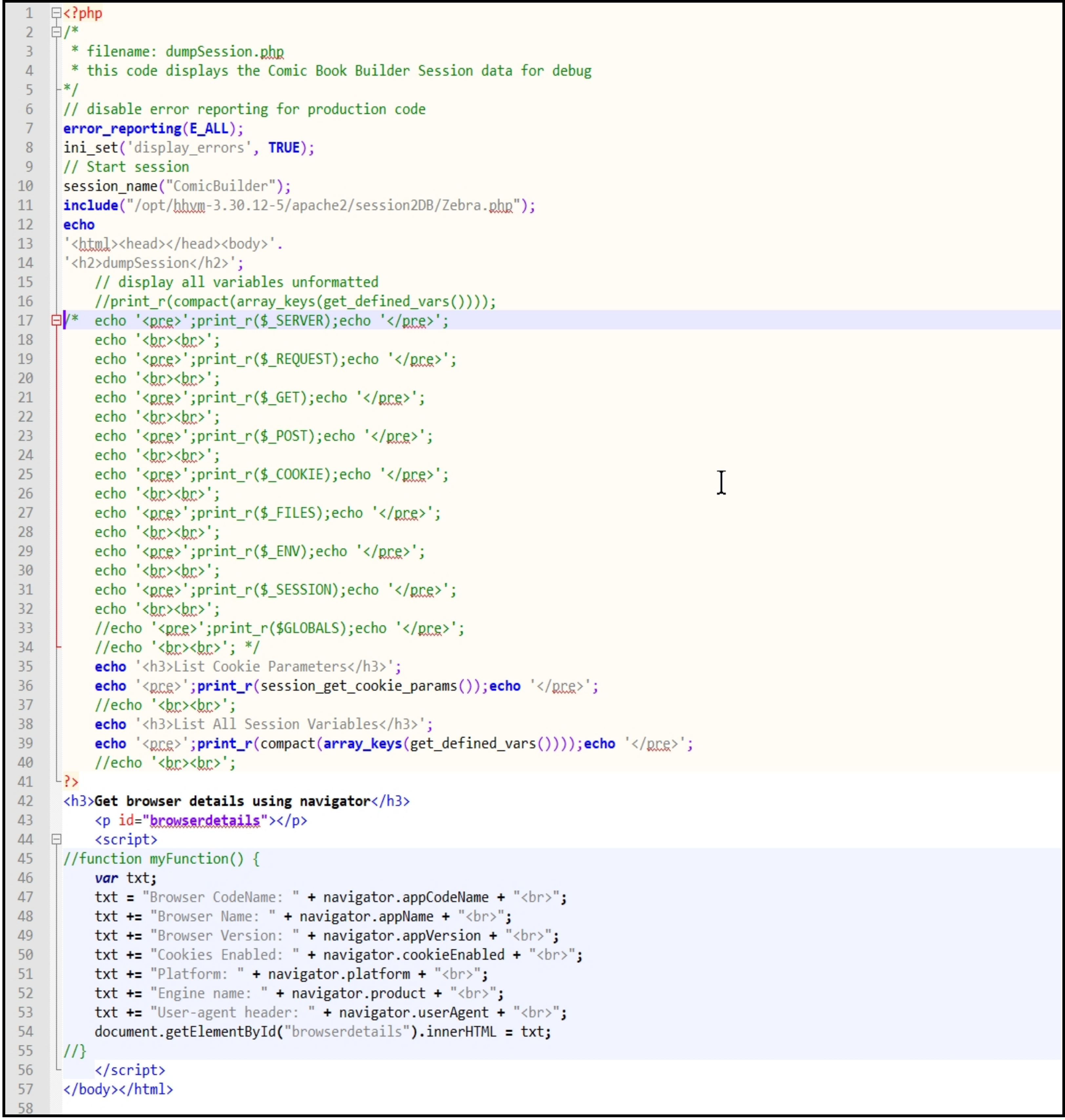
It is suggested that you read over the tutorial completely before you begin performing the operations described.

We need to control access to certain folders in such a way as to be consistent with security considerations and the operation of our code. The previous tutorial set these folders up as shown here.

```
root@BobosRevenge:/opt/hhvm-3.30.12-5/apache2# ll
total 8448
drwxr-xr-x 18 root root       4096 Oct 23 13:13 ./
drwxr-xr-x 20 root root       4096 Oct 16 20:52 ../
drwxrwsr-x  4 root www-pub    4096 Oct 23 13:13 Storybook/
drwxr-xr-x  2 root root       4096 Oct 16 20:49 bin/
-r-xr--r--  1 root root    8573582 Oct  3 10:09 bnconfig*
drwxr-xr-x  2 root root       4096 Oct 16 20:49 build/
drwxr-xr-x  2 root root       4096 Oct 16 20:51 cgi-bin/
drwxr-xr-x  5 root root       4096 Oct 17 12:52 conf/
drwxr-xr-x  3 root root       4096 Oct 16 20:49 error/
drwxrwsr-x  6 root www-pub    4096 Oct 23 17:37 htdocs/
drwxr-xr-x  3 root root       4096 Oct 16 20:49 icons/
drwxr-xr-x  2 root root       4096 Oct 16 20:49 include/
drwxrwsr-x  2 root www-pub    4096 Oct 23 13:05 includes/
drwxr-xr-x  2 root root       4096 Oct 23 20:55 logs/
drwxr-xr-x  2 root root       4096 Oct 16 20:49 modules/
drwxr-xr-x  2 root root       4096 Oct 16 20:49 scripts/
drwxrwsr-x  2 root www-pub    4096 Oct 23 13:10 session2DB/
drwxrwsr-x  2 root www-pub    4096 Oct 23 12:45 uploads/
drwxr-xr-x  3 root root       4096 Oct 16 20:49 var/
root@BobosRevenge:/opt/hhvm-3.30.12-5/apache2#
```

We will need to create or copy/paste a number of files into our stack. Copies of the files are included in this repo.

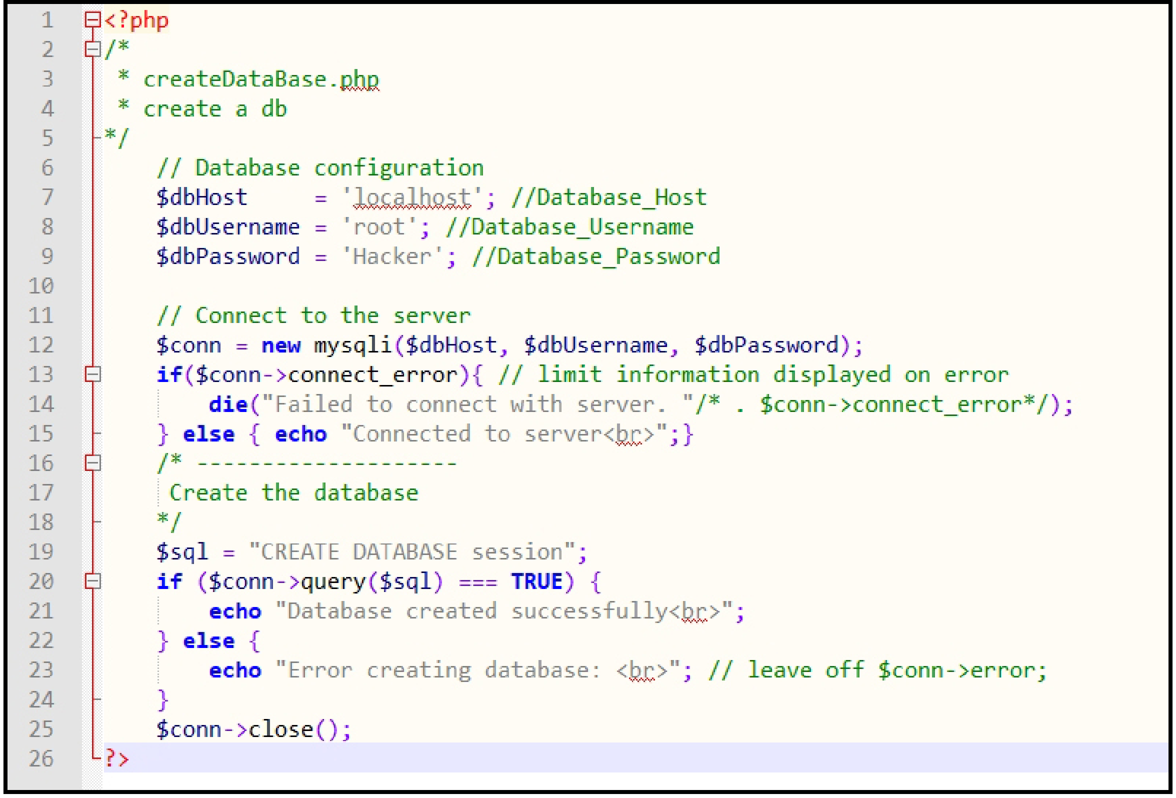
The first four files will go into our htdocs folder. The first file, dumpSession.php shown here will exercise our session manager functionality when we have it set up.



```
<?php
/*
 * filename: dumpSession.php
 * this code displays the Comic Book Builder Session data for debug
*/
// disable error reporting for production code
error_reporting(E_ALL);
ini_set('display_errors', TRUE);
// Start session
session_name("ComicBuilder");
include("/opt/hhvm-3.30.12-5/apache2/session2DB/Zebra.php");
echo
'<html><head></head><body>'.
'<h2>dumpSession</h2>';
    // display all variables unformatted
    //print_r(compact(array_keys(get_defined_vars())));
/*  echo '<pre>';print_r($_SERVER);echo '</pre>';
    echo '<br><br>';
    echo '<pre>';print_r($_REQUEST);echo '</pre>';
    echo '<br><br>';
    echo '<pre>';print_r($_GET);echo '</pre>';
    echo '<br><br>';
    echo '<pre>';print_r($_POST);echo '</pre>';
    echo '<br><br>';
    echo '<pre>';print_r($_COOKIE);echo '</pre>';
    echo '<br><br>';
    echo '<pre>';print_r($_FILES);echo '</pre>';
    echo '<br><br>';
    echo '<pre>';print_r($_ENV);echo '</pre>';
    echo '<br><br>';
    echo '<pre>';print_r($_SESSION);echo '</pre>';
    echo '<br><br>';
    //echo '<pre>';print_r($GLOBALS);echo '</pre>';
    //echo '<br><br>'; */
    echo '<h3>List Cookie Parameters</h3>';
    echo '<pre>';print_r(session_get_cookie_params());echo '</pre>';
    //echo '<br><br>';
    echo '<h3>List All Session Variables</h3>';
    echo '<pre>';print_r(compact(array_keys(get_defined_vars())));echo '</pre>';
    //echo '<br><br>';
?>
<h3>Get browser details using navigator</h3>
    <p id="browserdetails"></p>
    <script>
//function myFunction() {
    var txt;
    txt = "Browser CodeName: " + navigator.appCodeName + "<br>";
    txt += "Browser Name: " + navigator.appName + "<br>";
    txt += "Browser Version: " + navigator.appVersion + "<br>";
    txt += "Cookies Enabled: " + navigator.cookieEnabled + "<br>";
    txt += "Platform: " + navigator.platform + "<br>";
    txt += "Engine name: " + navigator.product + "<br>";
    txt += "User-agent header: " + navigator.userAgent + "<br>";
    document.getElementById("browserdetails").innerHTML = txt;
//}
    </script>
</body></html>
```

dumpSession_php.jpg

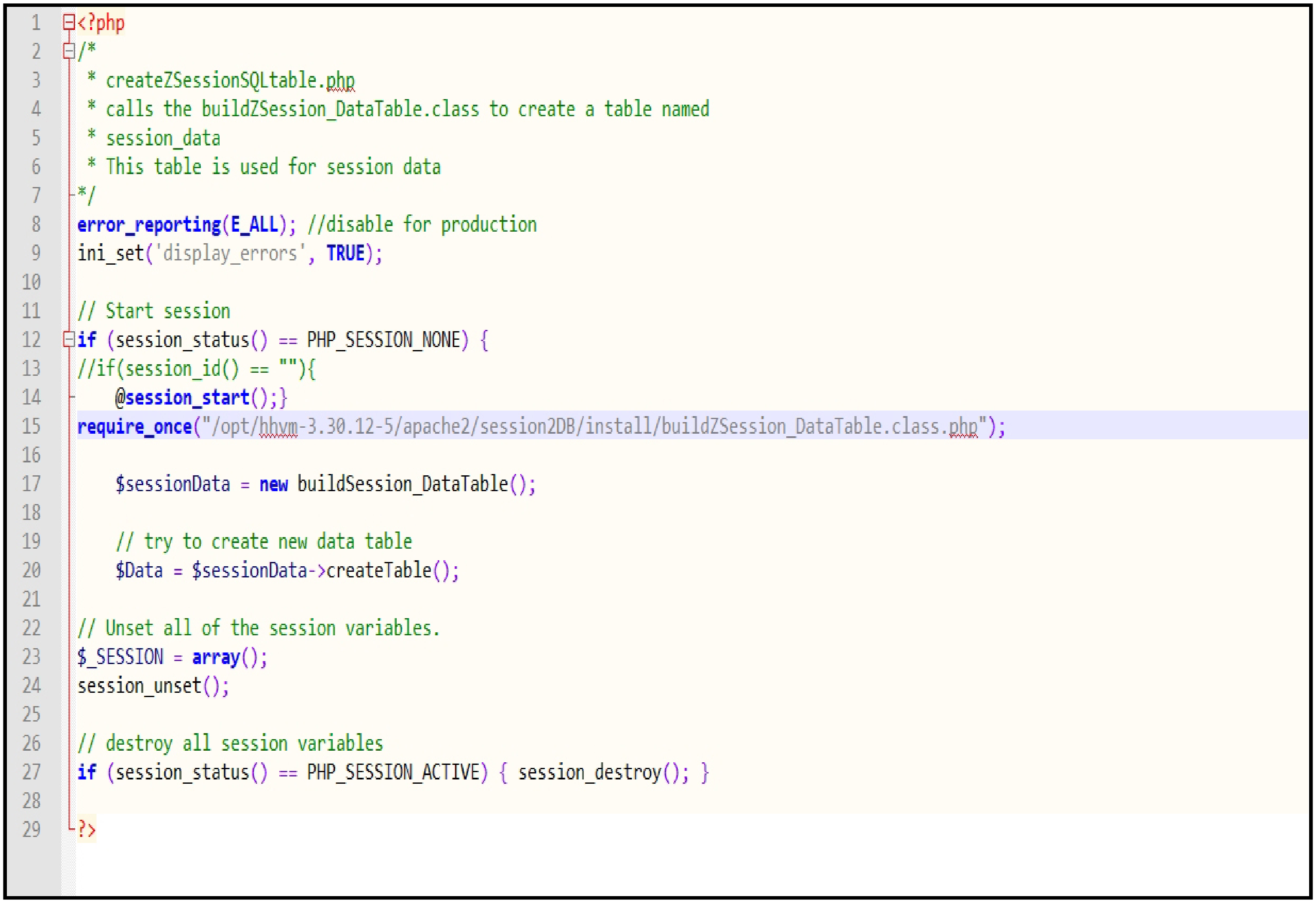
The second file, createZSessionDataBase.php shown here will set up the session manager database.



```php
<?php
/*
 * createDataBase.php
 * create a db
*/
    // Database configuration
    $dbHost     = 'localhost'; //Database_Host
    $dbUsername = 'root'; //Database_Username
    $dbPassword = 'Hacker'; //Database_Password

    // Connect to the server
    $conn = new mysqli($dbHost, $dbUsername, $dbPassword);
    if($conn->connect_error){ // limit information displayed on error
        die("Failed to connect with server. "/* . $conn->connect_error*/);
    } else { echo "Connected to server<br>";}
    /* --------------------
     Create the database
    */
    $sql = "CREATE DATABASE session";
    if ($conn->query($sql) === TRUE) {
        echo "Database created successfully<br>";
    } else {
        echo "Error creating database: <br>"; // leave off $conn->error;
    }
    $conn->close();
?>
```

createDataBase_php.jpg

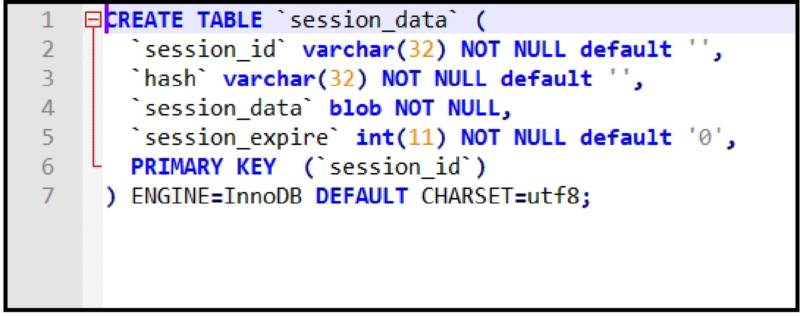
The third file, createZSessionSQLtable.php shown here will set up the session manager table in the database.



```php
<?php
/*
 * createZSessionSQLtable.php
 * calls the buildZSession_DataTable.class to create a table named
 * session_data
 * This table is used for session data
*/
error_reporting(E_ALL); //disable for production
ini_set('display_errors', TRUE);

// Start session
if (session_status() == PHP_SESSION_NONE) {
//if(session_id() == ""){
    @session_start();}
require_once("/opt/hhvm-3.30.12-5/apache2/session2DB/install/buildZSession_DataTable.class.php");

    $sessionData = new buildSession_DataTable();

    // try to create new data table
    $Data = $sessionData->createTable();

// Unset all of the session variables.
$_SESSION = array();
session_unset();

// destroy all session variables
if (session_status() == PHP_SESSION_ACTIVE) { session_destroy(); }

?>
```

createSessionSQLtable_php.jpg

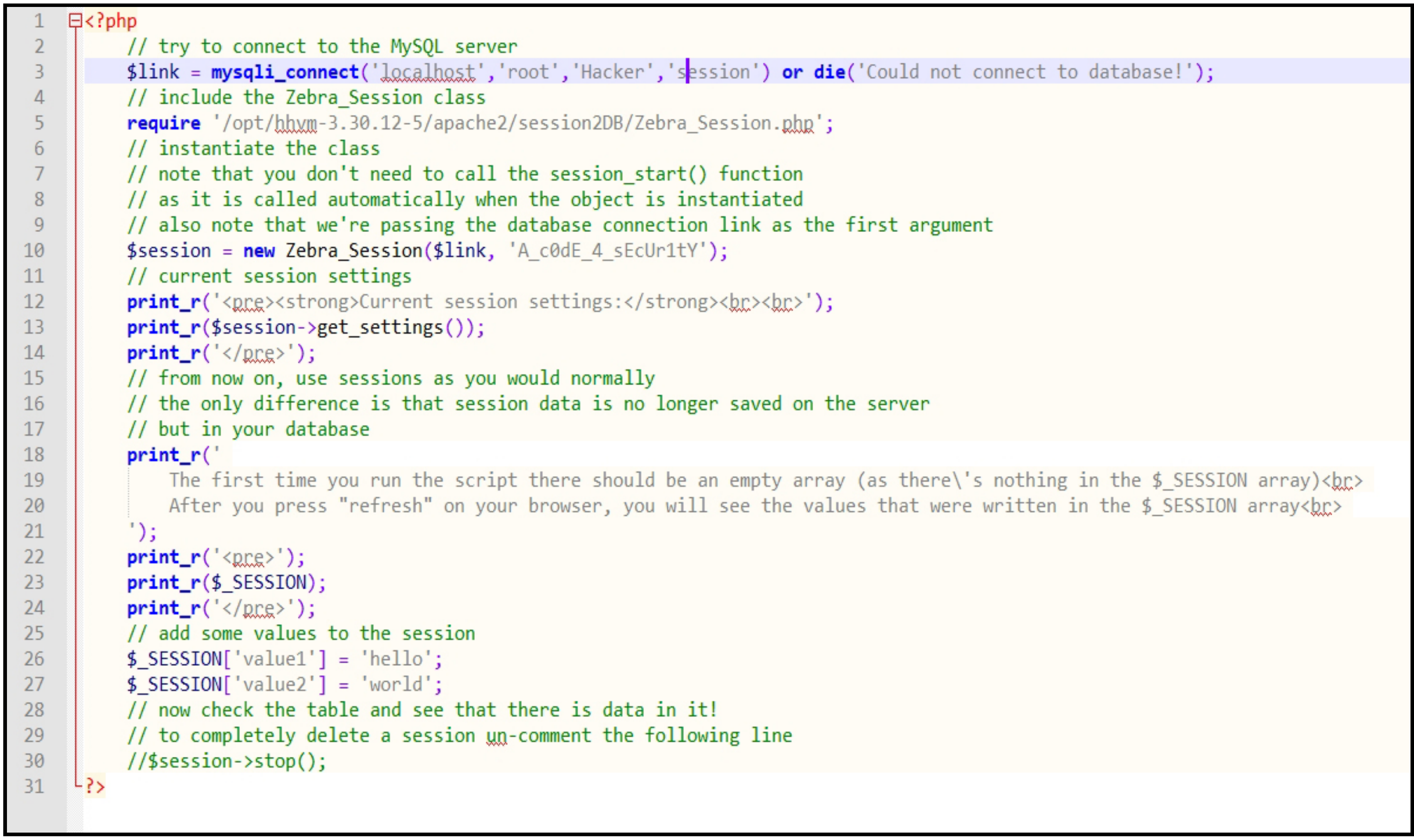
The structure of the table created by createZSessionSQLtable.php is shown here.



```
CREATE TABLE `session_data` (
  `session_id` varchar(32) NOT NULL default '',
  `hash` varchar(32) NOT NULL default '',
  `session_data` blob NOT NULL,
  `session_expire` int(11) NOT NULL default '0',
  PRIMARY KEY  (`session_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

SessionDataTableStructure.jpg

The fourth file, ZebraTest.php shown here will test our session manager functionality when we have it set up.

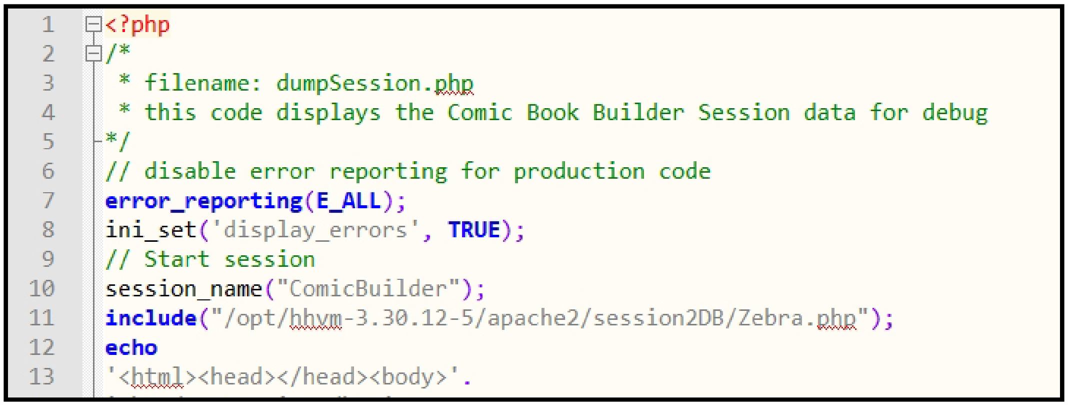


```
<?php
    // try to connect to the MySQL server
    $link = mysqli_connect('localhost','root','Hacker','session') or die('Could not connect to database!');
    // include the Zebra_Session class
    require '/opt/hhvm-3.30.12-5/apache2/session2DB/Zebra_Session.php';
    // instantiate the class
    // note that you don't need to call the session_start() function
    // as it is called automatically when the object is instantiated
    // also note that we're passing the database connection link as the first argument
    $session = new Zebra_Session($link, 'A_c0dE_4_sEcUr1tY');
    // current session settings
    print_r('<pre><strong>Current session settings:</strong><br><br>');
    print_r($session->get_settings());
    print_r('</pre>');
    // from now on, use sessions as you would normally
    // the only difference is that session data is no longer saved on the server
    // but in your database
    print_r('
        The first time you run the script there should be an empty array (as there\'s nothing in the $_SESSION array)<br>
        After you press "refresh" on your browser, you will see the values that were written in the $_SESSION array<br>
    ');
    print_r('<pre>');
    print_r($_SESSION);
    print_r('</pre>');
    // add some values to the session
    $_SESSION['value1'] = 'hello';
    $_SESSION['value2'] = 'world';
    // now check the table and see that there is data in it!
    // to completely delete a session un-comment the following line
    //$session->stop();
?>
```

ZebraTest_php.jpg

In addition to these files above in the htdocs folder, we need to create or copy/paste a number of files into our session2DB folder and into the install subfolder within it. Copies of the files are in the session2DB folder in this tutorial, along with a copy of the install subfolder and its files.

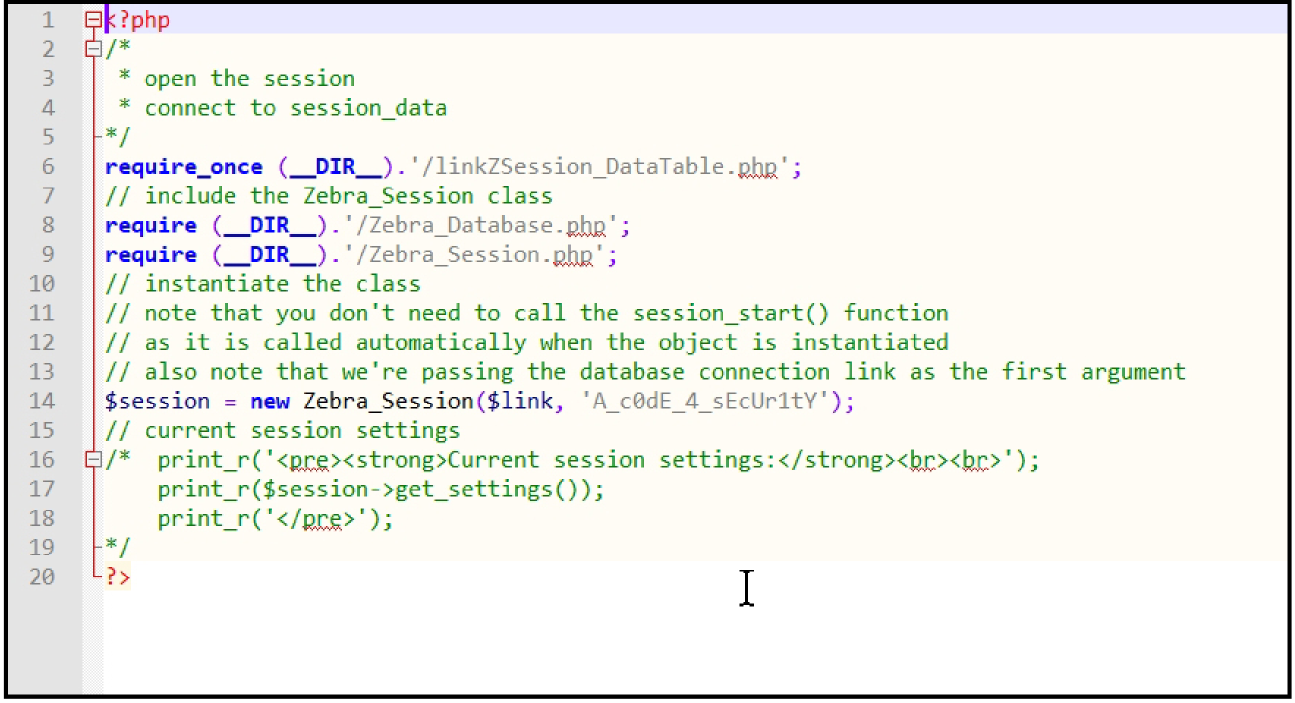
Besides the two main "executables", Zebra_Session.php and Zebra_Database.php, there are two other files of importance. One of these files, Zebra.php, is used to invoke the session manager at the beginning of our php programs as we may see here in our dumpSession code.



```
<?php
/*
 * filename: dumpSession.php
 * this code displays the Comic Book Builder Session data for debug
*/
// disable error reporting for production code
error_reporting(E_ALL);
ini_set('display_errors', TRUE);
// Start session
session_name("ComicBuilder");
include("/opt/hhvm-3.30.12-5/apache2/session2DB/Zebra.php");
echo
'<html><head></head><body>'.
```

startSessionHeaderForCode.jpg

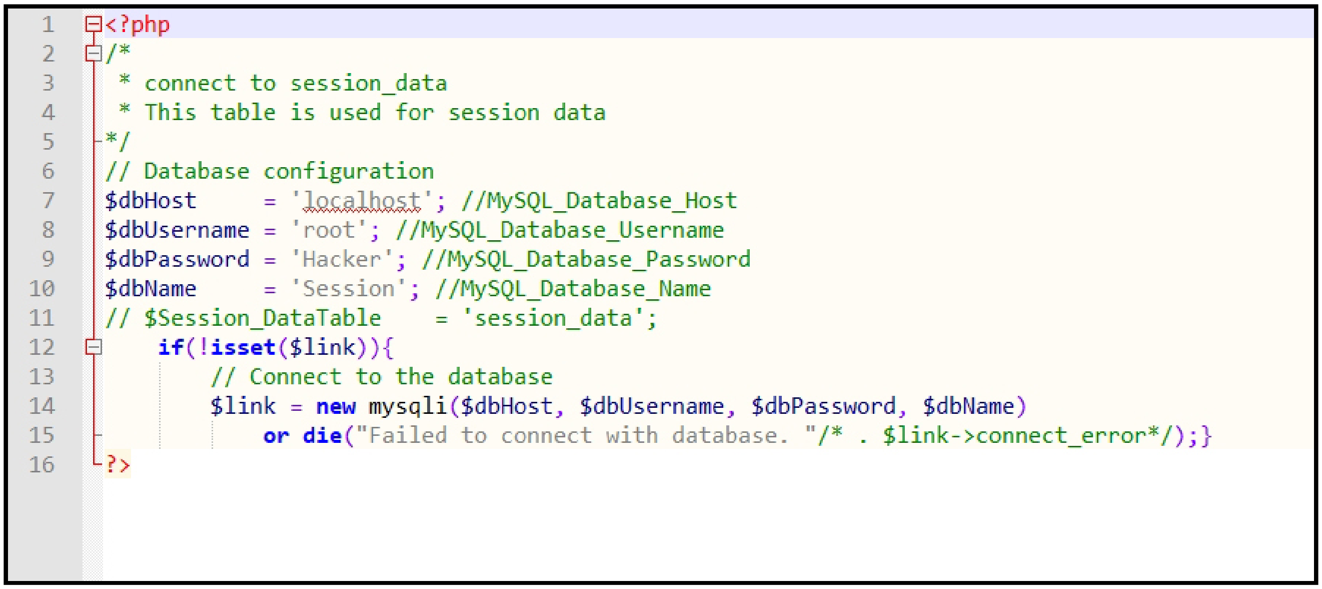
The contents of Zebra.php are shown here next.



```
<?php
/*
 * open the session
 * connect to session_data
*/
require_once (__DIR__).'/linkZSession_DataTable.php';
// include the Zebra_Session class
require (__DIR__).'/Zebra_Database.php';
require (__DIR__).'/Zebra_Session.php';
// instantiate the class
// note that you don't need to call the session_start() function
// as it is called automatically when the object is instantiated
// also note that we're passing the database connection link as the first argument
$session = new Zebra_Session($link, 'A_c0dE_4_sEcUr1tY');
// current session settings
/*  print_r('<pre><strong>Current session settings:</strong><br><br>');
    print_r($session->get_settings());
    print_r('</pre>');
*/
?>
```

Zebra_php.JPG

Particularly note the reference to the file linkZSession_DataTable.php which contains our database connection parameters as shown here next.



```
<?php
/*
 * connect to session_data
 * This table is used for session data
*/
// Database configuration
$dbHost     = 'localhost'; //MySQL_Database_Host
$dbUsername = 'root'; //MySQL_Database_Username
$dbPassword = 'Hacker'; //MySQL_Database_Password
$dbName     = 'Session'; //MySQL_Database_Name
// $Session_DataTable    = 'session_data';
    if(!isset($link)){
        // Connect to the database
        $link = new mysqli($dbHost, $dbUsername, $dbPassword, $dbName)
            or die("Failed to connect with database. "/* . $link->connect_error*/);}
?>
```

linkZSession_DataTable_php.JPG

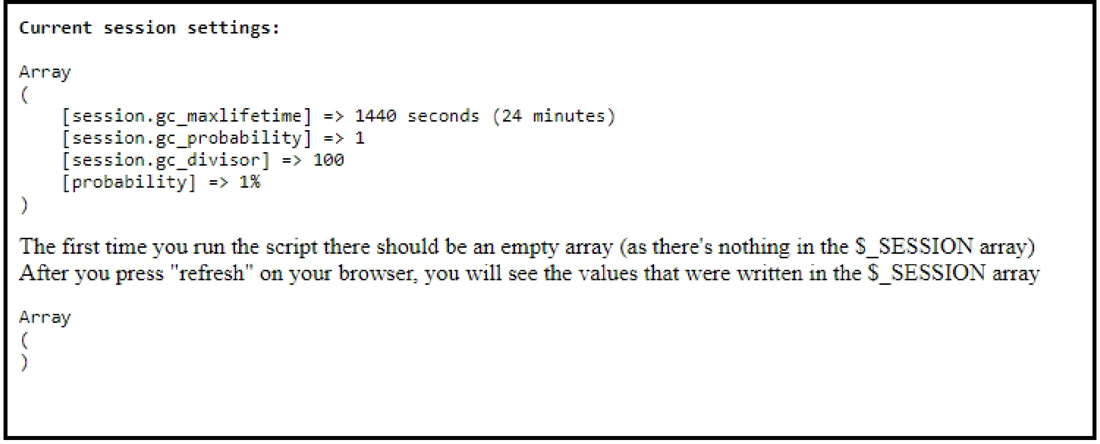
To use the database session manager simply replace the familiar session_start with an include statement like this, along with such options as the session_name value, so instead of

```
//Start session
session_name("ComicBuilder");
session_start;
```

we will now use

```
// Start session
session_name("ComicBuilder");
include("/opt/hhvm-3.30.12-5/apache2/session2DB/Zebra.php");
```

If we run ZebraTest.php from our browser at the URL 127.0.0.1:8080/ZebraTest.php we will see this first screen shown here.



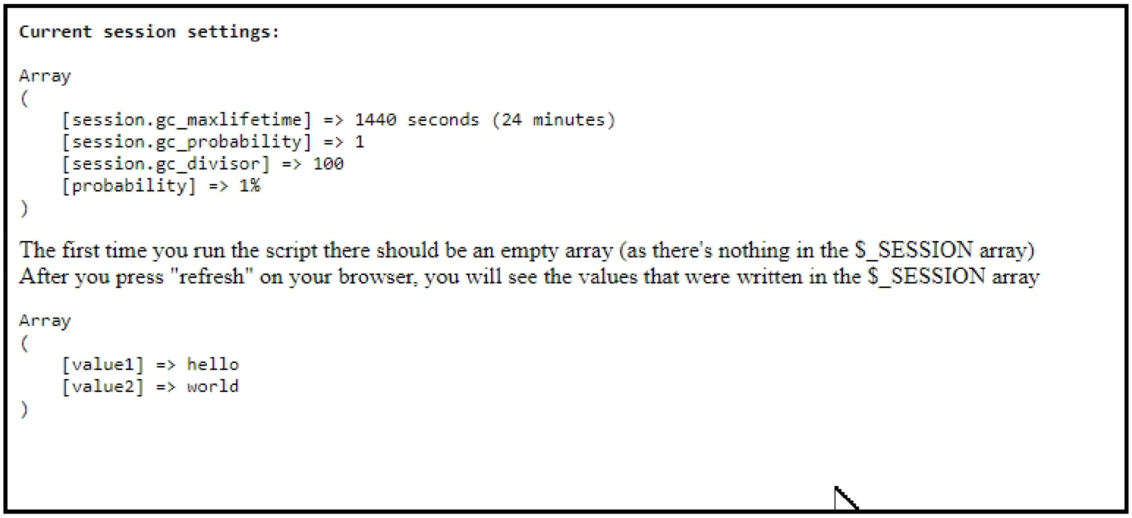
```
Current session settings:

Array
(
    [session.gc_maxlifetime] => 1440 seconds (24 minutes)
    [session.gc_probability] => 1
    [session.gc_divisor] => 100
    [probability] => 1%
)
```

The first time you run the script there should be an empty array (as there's nothing in the $_SESSION array)
After you press "refresh" on your browser, you will see the values that were written in the $_SESSION array

```
Array
(
)
```

ZebraTestScreen1.JPG

If we refresh the browser as this first screen suggests we will then see this second screen.



```
Current session settings:

Array
(
    [session.gc_maxlifetime] => 1440 seconds (24 minutes)
    [session.gc_probability] => 1
    [session.gc_divisor] => 100
    [probability] => 1%
)
```

The first time you run the script there should be an empty array (as there's nothing in the $_SESSION array)
After you press "refresh" on your browser, you will see the values that were written in the $_SESSION array

```
Array
(
    [value1] => hello
    [value2] => world
)
```

ZebraTestScreen2.JPG

More information can be obtained by running the dumpSession.php script mentioned above which results in a fairly large amount of data output so it is included as a text file dumpSessionOutput.txt. For extra credit you can use these following lines in a php scrpt modification to our index.php file in the htdocs folder. (A line break was inserted here after the colon in the branch section of the defined statement for the PDF formatting.) I named this code index2.php. The phpinfo results will then reflect the session handler usage.

```
// Start session
session_name("ComicBuilder");
include("/opt/hhvm-3.30.12-5/apache2/session2DB/Zebra.php");
echo defined('HHVM_VERSION') ? 'Great! You are Using HHVM' :
 'Sorry! Not using HHVM';
phpinfo();
```

# We now have a database session handler for our PHP code installed on our LAMH stack.

Another thing we can do is to download and install the mySQL Workbench tool from the mySQL developer site. You will need to provide the server name localhost or the ip (I used 127.0.0.1), the default user name is root, and the password is Hacker. The port number is the usual 3306. Once it is set up you can connect from your Windows machine and obtain a screen like this shown next. (I made a couple of efforts to install phpmyadmin on the stack but was unsuccessful at that effort.)

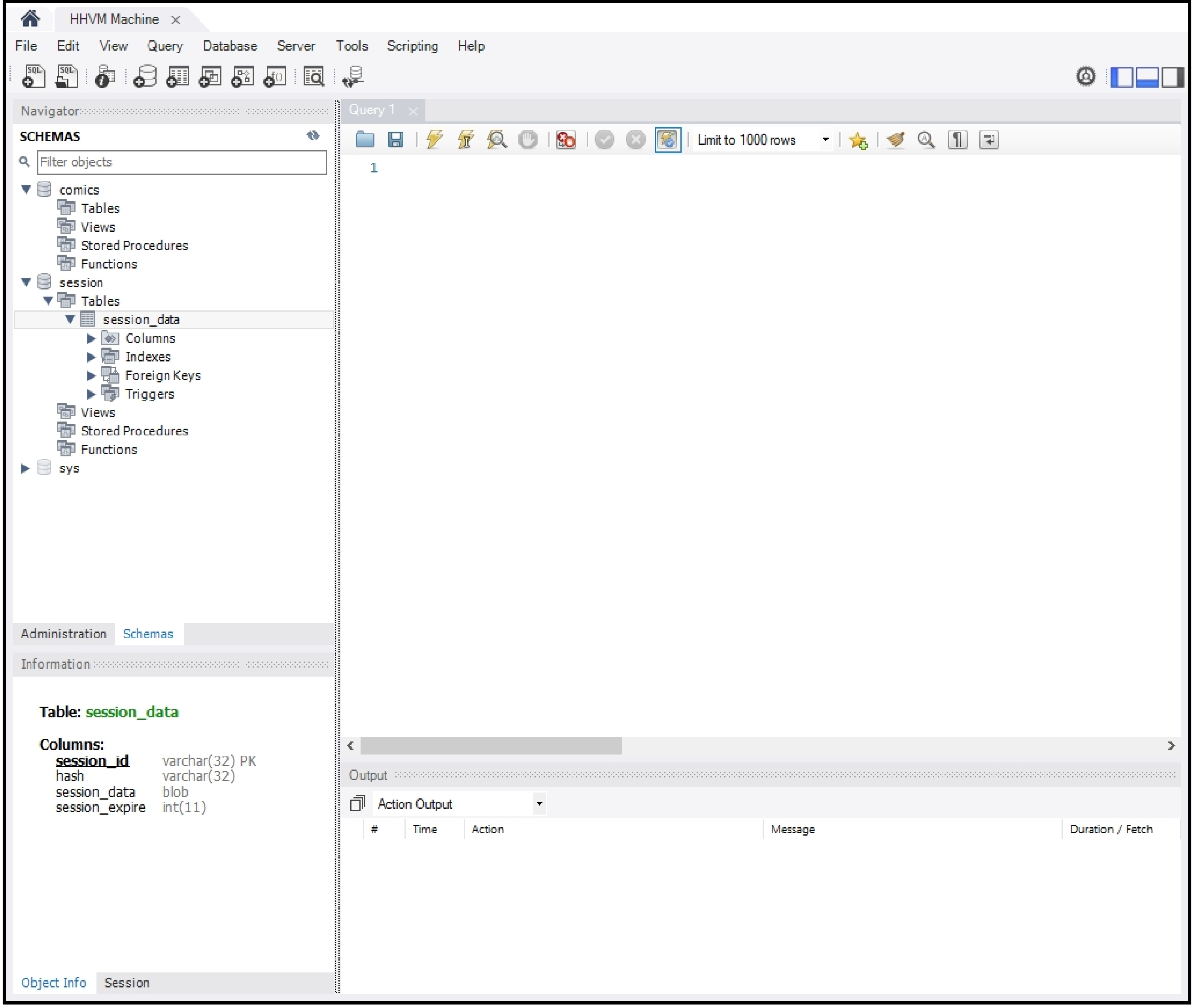


**Table:** session_data

**Columns:**

```
session_id      varchar(32) PK
hash            varchar(32)
session_data    blob
session_expire  int(11)
```

mySQLWorkbenchScreenshot.JPG

The final step we should take is to execute these commands in our Windows Administrator Command prompt to back up the changes we have made.

```
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\WINDOWS\system32>wsl --list --all
Windows Subsystem for Linux Distributions:
Ubuntu-18.04 (Default)
C:\WINDOWS\system32>wsl --export Ubuntu-18.04 D:/UbuntuHHVMbase.tar
C:\WINDOWS\system32>
```

Next time we'll use more of this files-folders-users hierarchy along with some database in our LAMH stack and web sites.