Building an E-mini Trading System Using PHP and Advanced MySQL Queries

(2009-11-10) - Contributed by Codex-M

This article shows illustrative examples of how PHP and some advanced MySQL queries can be used to build an online trading system. For simplicity, we will be featuring one of the most common stock indexes: the S&P 500 index.

This is also called E-mini S&P 500 futures, one of the most commonly traded equities/futures.

The Design

As of September 2009, there was no currently working trading system on the Internet powered by PHP and MySQL despite the platform's popularity among open source communities.

A trading system is used to enter and exit a trade in a stock or futures market. Although in exiting a trade to grab profits and control losses, money management is a more useful tool than a trading system.

To create a fully working trading system, you need the following processes:

The raw data should be taken from a reputable source for S&P 500 index data. In this article, we will be using Yahoo! Finance data. The export file type is mostly .csv and Excel, so you first need to export the raw data to the MySQL database. (See details below)

The heart of the process is the trading system. It is a method for analyzing raw data and making judgments. Every trader has his or her own trading system to follow, but in this example, we will use the following rules:

Rule 1: Define Delta as the difference between “50 days moving average “and “200 moving average.” If the delta is positive, it indicates a bullish trend, otherwise it's bearish. You can read more about moving averages.

Rule 2: Define % Strength of the trend as Delta/MA 200. This quantifies the strength of the trend. Apply historical data (define limits) for judgment (computation of limits is out of scope for this tutorial). This is more like a trend confirmation.

Use PHP to display the data in HTML tables while using MySQL advance queries (computing average and getting index data) to do some calculations of the moving average.

{mospagebreak title=Implementation, steps 1 through 5}

Step 1: Download raw S&P 500 index data from http://finance.yahoo.com/q/hp?s=%5EGSPC and format the Excel table to look like the image below (delete unneeded columns like trading volumes and add the new column “entry”):

Your MySQL table should look like the one above. However, the above screen shot is still in .csv format

Step 2: Convert the .csv to .sql and upload to your MySQL server. You can read an article that explains how to convert your Excel worksheet.

Step 3: It's time to write the PHP script. First we connect to the database using the script below:

```
//connect to mysql database

$username = "your mysql username";

$password = "your mysql password";

$hostname = "your mysql hostname";

$database = "your mysql database";

$dbhandle = mysql_connect($hostname, $username, $password)

or die("Unable to connect to MySQL");

//select a database to work with

$selected = mysql_select_db($database,$dbhandle)

or die("Could not select $database");
```

Step 4: In order to know how many rows are in the database, which is a value that will be used in our computation, we will query MySQL:

```
//count the number of rows in the database including the latest entry

$result1 = mysql_query("SELECT * FROM `sp500`")

or die(mysql_error());

// store the record of the "example" table into $row

$row1 = mysql_num_rows($result1)

or die("Invalid query: " . mysql_error());

// Print out the contents of the entry

$numberofentries = $row1;
```

The $numberofentries contains the maximum number of rows in the MySQL table (in the above screenshot it is 15020). The data in the MySQL table will be arranged from the latest entries all the way down to old entries. Refer to the screenshot above.

The mysql_num_rows will be used to count the number of rows in the table SP500.

Step 5: Compute the last entry to be shown in the HTML table. Since we will be showing only the latest 50 rows, the following will be used:

```
$lastentry = $numberofentries -50 +1;
```

This means that, for example, we have 15020 entries in the table; we would like to show them starting from Entry # 15020 to (15020-50+1) or Entry # 14971. This value, as well as the maximum entry above, will change as trading days goes by.

{mospagebreak title=Implementation, steps 6 through 13}

Step 6: Do a MySQL query to extract the latest 50 days of entries.

```
$result2 = mysql_query("SELECT * FROM `sp500` WHERE `entry`>='$lastentry' AND `entry`<='$numberofentries' ORDER BY `entry` DESC")

or die(mysql_error());
```

The tricky part is to sort the resulted queries by descending order. This will ensure that output results are sorted from the newest to oldest entries.

Step 7: Define the limits for the 200-day moving average and 50-day moving average:

```
/Define limits for 200 day moving average

$lowerlimit= $numberofentries - 200 + 1;

$upperlimit= $numberofentries;

//Define limits for 50 day moving average

$lowerlimit50= $numberofentries - 50 + 1;

$upperlimit= $numberofentries;
```

Step 8: Define the MySQL query to calculate the 200-day moving average.

```
$result3 = mysql_query("SELECT avg(close) from `sp500` WHERE `entry`>='$lowerlimit' AND `entry`<='$upperlimit'")

or die(mysql_error());

// store the record of the "example" table into $row

$row3 = mysql_fetch_array($result3)
```

```
or die("Invalid query: " . mysql_error());

// Print out the contents of the entry

$ma200 = $row3['avg(close)'];
```

Step 9: Define the MySQL query to calculate the 50-day moving average.

```
$result4 = mysql_query("SELECT avg(close) from `sp500` WHERE `entry`>='$lowerlimit50' AND `entry`<='$upperlimit'")

or die(mysql_error());

// store the record of the "example" table into $row

$row4 = mysql_fetch_array($result4)

or die("Invalid query: " . mysql_error());

// Print out the contents of the entry

$ma50 = $row4['avg(close)'];
```

Step 10: Assign to variables and round numbers for easier HTML display.

```
$w=round($ma200,2);

$x=round($ma50,2);
```

Step 11: Compute DELTA and round results.

```
$y=round(($ma50-$ma200),2);
```

Step 12: Compute %STRENGTH and round results.

```
$z=round((((($ma50-$ma200)/$ma200)*100),3);
```

Step 13: Create the PHP script to make “recommendations.”

```
if ($z >= 8.51) {

$recommendation = 'above SATURATED BULL TREND( VERY HIGH RISK BUYING)';
```

```
}

elseif ( $z >=5.1 && $z <=8.5) {

$recommendation = 'CONFIRMED BULL TREND (HIGH RISK BUYING)';

}

elseif ( $z >=2 && $z <=5) {

$recommendation = 'CONFIRMED BULL TREND (LOW RISK BUYING)';

}

elseif ( $z >=0 && $z <=1.99) {

$recommendation = 'UNCONFIRMED BULL TREND';

}

elseif ( $z <=0 && $z >=-4.99) {

$recommendation = 'UNCONFIRMED BEAR TREND';

}

elseif ( $z <=-5 && $z >=-8) {

$recommendation = 'CONFIRMED BEAR TREND (LOW RISK SHORT)';

}

elseif ( $z <=-8.1 ) {

$recommendation = 'SATURATED BEAR TREND( VERY HIGH RISK SHORT)';

}
```

Note: threshold values were taken from historical records of S&P. These computations are out of the scope of this tutorial.

{mospagebreak title=Implementation, steps 14 and 15}

Step 14: Combine all queries above and computations. Then print to an HTML table using a WHILE loop:

```
if (mysql_num_rows($result2) >0)

{

//print analysis

//echo '<table width=100% cellpadding=10 cellspacing=10 border=1>';

echo '<table width=100% border=1>';

echo '<tr><td>Entry</td><td>Date</td><td>Open
</td><td>High</td><td>Low</td><td>Close</td>
<td>MA200</td><td>MA50</td><td>Delta</td>
<td>%Strength</td><td>Recommendation</td></tr>';

while ($row2= mysql_fetch_row($result2))

{

$result3 = mysql_query("SELECT avg(close) from `sp500` WHERE `entry`>='$lowerlimit' AND `entry`<='$upperlimit'")

or die(mysql_error());

// store the record of the "example" table into $row

$row3 = mysql_fetch_array($result3)

or die("Invalid query: " . mysql_error());

// Print out the contents of the entry

$ma200 = $row3['avg(close)'];

$result4 = mysql_query("SELECT avg(close) from `sp500` WHERE `entry`>='$lowerlimit50' AND `entry`<='$upperlimit'")

or die(mysql_error());

// store the record of the "example" table into $row

$row4 = mysql_fetch_array($result4)

or die("Invalid query: " . mysql_error());

// Print out the contents of the entry
```

```
$ma50 = $row4['avg(close)'];

$w=round($ma200,2);

$x=round($ma50,2);

$y=round(($ma50-$ma200),2);

$z=round((((($ma50-$ma200)/$ma200)*100),3);

if ($z >= 8.51) {

$recommendation = 'above SATURATED BULL TREND( VERY HIGH RISK BUYING)';

}

elseif ( $z >=5.1 && $z <=8.5) {

$recommendation = 'CONFIRMED BULL TREND (HIGH RISK BUYING)';

}

elseif ( $z >=2 && $z <=5) {

$recommendation = 'CONFIRMED BULL TREND (LOW RISK BUYING)';

}

elseif ( $z >=0 && $z <=1.99) {

$recommendation = 'UNCONFIRMED BULL TREND';

}

elseif ( $z <=0 && $z >=-4.99) {

$recommendation = 'UNCONFIRMED BEAR TREND';

}

elseif ( $z <=-5 && $z >=-8) {
```

```
$recommendation = 'CONFIRMED BEAR TREND (LOW RISK SHORT)';

}

elseif ( $z <=-8.1 ) {

$recommendation = 'SATURATED BEAR TREND( VERY HIGH RISK SHORT)';

}

$upperlimit--;

$lowerlimit50--;

$lowerlimit--;

echo '<tr>';

echo '<td>'.$row2[0].'</td>';

echo '<td>'.$row2[1].'</td>';

echo '<td>'.$row2[2].'</td>';

echo '<td>'.$row2[3].'</td>';

echo '<td>'.$row2[4].'</td>';

echo '<td>'.$row2[5].'</td>';

echo '<td>'.$w.'</td>';

echo '<td>'.$x.'</td>';

echo '<td>'.$y.'</td>';

echo '<td>'.$z.'</td>';

echo '<td>'.$recommendation.'</td>';

echo '</tr>';

}
```

```
echo '</table>';

}

else

{

echo '<br />No rows found.<br />';

}
```

Step 15: Free the MySQL results and close the database connection.

```
mysql_free_result($result1);

mysql_free_result($result2);

mysql_free_result($result3);

mysql_free_result($result4);
mysql_close($dbhandle);
```

This is a fairly complicated application. You can see it in action here: http://www.php-developer.org/SP500historicaldataanalysis.php

Or download the complete PHP script.