

# **Lies, Damned Lies, Statistics and SQL**

by  
Peter Lavin

December 17, 2003

## **Introduction**

When I read about the Developer Shed December Giveaway Contest in the most recent newsletter a thought occurred to me. Given the nature of the Developer Shed Network of sites, there should be an objective, computer-driven way of determining the winner. To remove all question of subjective bias in determining the winner I would like to suggest an algorithm to do just this. I will use the data provided by the hits statistics on the Dev Articles site. This should prove to be an ideal way to achieve an unprejudiced solution.

This article will show how these hit statistics can quickly be imported into an Access database and manipulated using SQL to objectively determine a winner. If you wish you may perform the steps yourself or just read along.

## **Finding the Data**

If you go to the Dev Articles site and look at the left sidebar you'll find all the category descriptions followed by the number of articles in that category. Click on "PHP" to navigate to the most popular subject. You'll be presented with a list of the most recent submissions. Scroll to the bottom of the page, find and click the link to the full list of articles. This is where we will find detailed information about each submission.

Again the articles are sorted by the most recent submission – not a format that is particularly useful to us at the moment. You could probably eyeball the list and form a fairly accurate opinion of the most popular and most prolific authors but we want to be objective and precise so we're going to import this data into a database. There's far too much data here to even contemplate manual data entry. Let's see how we might import it with the least effort.

## **Capturing the Data**

Select all of the articles and related information on the PHP page, avoiding the table header, and copy them to the clipboard using menu options or simply by simultaneously pressing "Control" and "c". We're going to use Access so I'll assume you are running Windows. Open WordPad – make sure you don't use Notepad. Paste the contents of the clipboard into WordPad. Looks like quite a mess doesn't it? I'll bet you're wondering how this can be any use to us – have faith and you'll see shortly.

We are now going to save this file in text format so choose "Save As" from the "File" menu option. You should be presented with a file dialogue box. Choose

“text document” as the file type and change the file name to “hits.txt”. Make sure you use the “.txt” extension. Access is picky about extensions when importing files and misbehaves if it doesn’t get its own way.

Before proceeding let’s have a look at the “hits.txt” file using Word. Right click the file name and open the “Send To” option. If you don’t find “MS Word“ here you can add it by simply putting a shortcut in the “Send To” directory. If you don’t want to do this just open “Word” and open the file, making sure that you select the right file type from the “File Type” dropdown box.

It doesn’t look quite as funny as it did previously – in fact it looks rather commonplace. To understand what’s going on the formatting marks need to be displayed. This is done from the “Tools” menu, by clicking “Options”, and then the “View” tab. Under “Formatting marks”, check the “All” box. Now you should see the text separated by “arrows” and funny looking backward “P’s”. You’re looking at tabs and carriage returns. In order for Access to import a text file it needs to be able to tell where the columns and rows end. The tabs will separate columns and the carriage returns will separate rows.

Your first line of text should look like the following:

```
→ 13·Dec· → Generating·Images·on·the·Fly·With·PHP·  
<http://www.devarticles.com/c/a/PHP/Generating_Images_on_the_Fly_With_PHP/>  
→ Divyesh·Jariwala→1095→¶
```

If it does then everything should probably work just fine. Make sure you don’t resave in Word format.

## Into the Database

We are now ready to convert our text file into a database table. Access is not an industrial-strength database like Oracle or DB2 and for this reason it is sometimes maligned. However that may be, for the job we are about to do I wouldn’t want any other database.

Open Access and create a new database. I’ve called mine “objectivewinner.mdb” but choose whatever name you like. If you already know how to import text files into a database you can skip this section and move on to the section entitled “More Data”. For the others here’s how it’s done.

Using the “File” menu option choose “Get External data “ and “Import”. Make sure you choose the “Text Files” file type and click on “hits.txt”. The “Import Text Wizard” opens and makes your job very easy. Choose the defaults until Access asks for a primary key. Choose “No primary Key”. Continue selecting the defaults through to the end.

When Access has finished importing it will report that there were errors. For our purposes, these are not important, so just ignore them. You should now have a table in your database called “hits”. We’re going to clean it up in a minute but let’s import some more data first.

## More Data

To ensure the objectivity of our exercise let’s *randomly* choose two other categories to import, namely the “Java” articles and the “MySQL” articles. (Be assured that the fact that the author of this article has published in both these categories has no bearing whatsoever.) Repeat the steps above to import these categories but make one exception, - choose to import your data into the existing table “hits”.

## Clean up the Database

Open the table “hits” in design view and delete “Field1” and “Field2”. These fields are not used so nothing will be lost. Rename the remaining fields “Title”, Author” and “Hits” in that order. We’re now ready to query the database and very soon we’ll have the name of December’s winner.

## The Contest SQL

The algorithm that will unlock the puzzle of December’s winner is shown below:

```
SELECT TOP 1 [Author], SUM([Hits])
AS SumOfHits, Count([author])
AS numarticles, Round((Sum([hits])/Count([author])))
AS avghits
FROM hits
GROUP BY [Author]
HAVING Count([author]) > 2
ORDER BY Round(Sum([hits])/Count([author])) DESC
```

Cut and paste this code into an SQL query window but don’t run it just yet. Let’s first say a few brief words about it.

Most of you are probably familiar with the common SQL keyword “SELECT” so let’s not dwell on it. The predicate “TOP 1” will ensure that only one record is returned – after all we are really only interested in the winner. Next comes the list of the fields to be retrieved. Obviously we want to include the Author field – we are going to need the winner’s name. “Sum([hits])” is an aggregate function that does exactly what you might think – namely it totals the “hits” field. Following this

is the keyword “AS” which will allow us to give a meaningful alias to this aggregate function. We’ve chosen the name “SumofHits” not too original perhaps but descriptive of the value returned.

After this, the “Round” function will ensure that no fractional values are returned. The formula inside the “Round” function will give us the average number of hits by author. This will be one of our *unprejudiced* criteria for determining the winner.

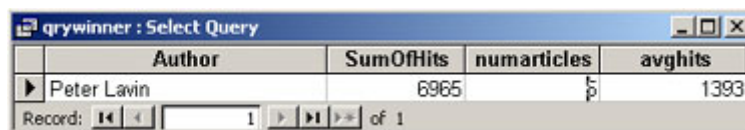
Where you have a select statement there is always a “FROM” clause lurking nearby. Since we only have one table in our database the table we’re selecting from is “hits”.

Whenever there is an aggregate function in a select statement we must group by any simple fields that appear in that select statement. Hence, we have “GROUP BY [Author]”. “HAVING” clauses are simply the replacement for “WHERE” clauses when a “GROUP BY” has been used. The expression “Count([author]) > 2” will ensure that our results are not skewed by statistical anomalies, further enhancing the *objectivity* of our results

Finally the “ORDER BY” clause will arrange the records in descending order starting with the author with the best average number of hits.

## Query Results

I don’t know if you can imagine my surprise when I saw the results returned by our query. If I was simply to report them you might not believe me so I’m going to show you them using a screen capture.



Author	SumOfHits	numarticles	avghits
Peter Lavin	6965	5	1393

In cases such as this, a more quarrelsome audience might question the credibility of our methods. Fortunately, this description does not apply to the readers of the Developer Shed Network especially at this time of year when they want to stay off the naughty list, and, frankly, given the thorough efforts at objectivity, I don’t think that there can be any reasonable grounds for dispute.

## Conclusion

We were able to quickly retrieve data from the Dev Articles site and save it as a text file with tab-separated values. This enabled us to import the file into a table

in Access. Querying this data using SQL statements allowed us to arrive at an objective determination of the December contest winner.

Finally, to the prize committee I would like to say “Thank you. Don’t wrap it I’m not putting it under the tree.”

## **About the Author**



Peter Lavin runs a Web Design/Development firm in Toronto, Canada. For more information visit <http://www.softcoded.com/>. He may be reached at [peterlavin@sympatico.ca](mailto:peterlavin@sympatico.ca).