# Dublin Business School

# Higher Diploma in Science in Data Analytics

## Module Programming for Big Data

## Assignment: CA-5 Perform Analytics on a 5,000 entry dataset

### Name: Alan Cleary

### Student Number: 10369236

## Reading of DATA

After the code was completed and the cleaned data of the git commits was written as a CSV, it was opened in Excel. The Instructions for this assignment stated that there would be 422 objects in the list that would be investigated however when the file was opened in Excel the File had 427 lines including a header. After some investigation it was discovered that excel splits some lines into two when the text exceeds a certain limit, which it did in this case. Further exploration provided a solution which was to open the file in Excel using the “From Text” option in the “data” ribbon. This method of opening the data truncated the data on the 3 occasions when the data was being split into 2 lines leaving the File with 423 lines (422 objects).

## Interesting Pieces of Information

The next requirement in this assignment was to come up with 3 interesting statistical pieces of information. This was done by exporting the CSV file to Microsoft Access and then querying the data using SQL.

### 1: The Total Number of commits made by each Author

The first query was written to analyse home many revisions had been made by each author.

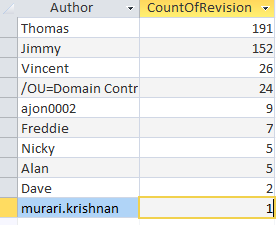
#### SQL:

SELECT Changes.Author, Count(Changes.Revision) AS CountOfRevision

FROM Changes

GROUP BY Changes.Author

ORDER BY Count(Changes.Revision) DESC;



As can be seen in the above image over the time period in question, Thomas and Jimmy produced over 80% of the total commits.

### 2: Max number of lines committed by an Author

After finding out which authors committed the most amount of code it was decided to find out each authors maximum lines of code committed at once.

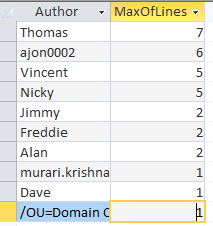
#### SQL:

SELECT Changes.Author, Max(Changes.[Lines Changed]) AS [MaxOfLines Changed]

FROM Changes

GROUP BY Changes.Author

ORDER BY Max(Changes.[Lines Changed]) DESC;



As can be seen above all the authors are committing less than 10 lines of code at a time.

### 3: Average number of lines per commit

Because the maximum lines of code committed was so few by each author, the next query decided on was what was the average number of lines committed in total on all query’s.

#### SQL:

SELECT Avg(Changes.[Lines Changed]) AS [AvgOfLines Changed]

FROM Changes;



As can be seen after running the above SQL query the Average number of lines committed in the file is 1.315 lines.