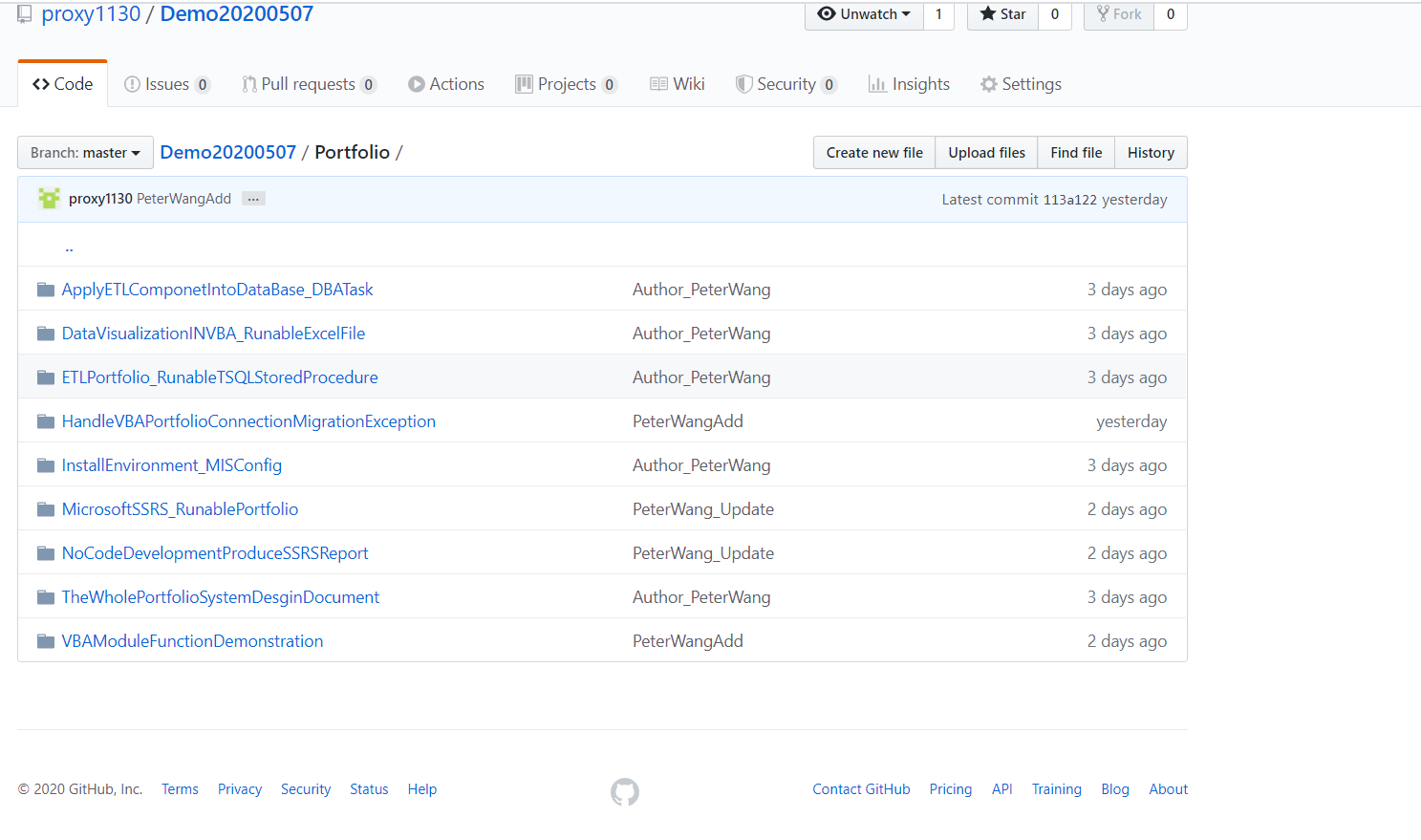
1. 為何如下圖所示，會有這個GitHub repository 所存放相關成品於GitHub呢?

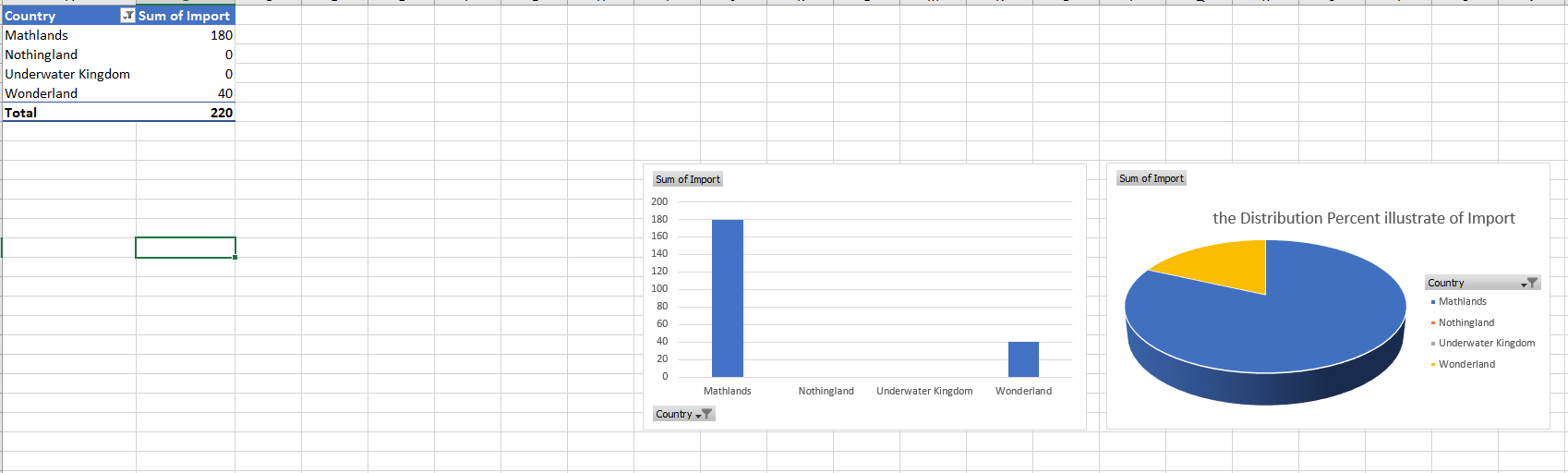
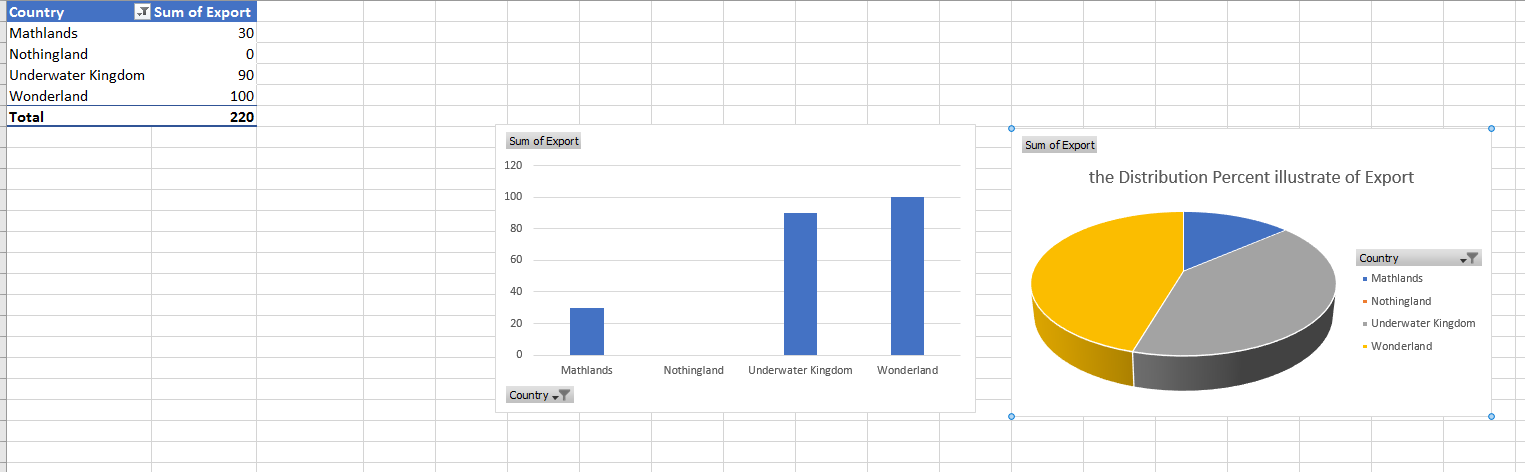


主要是因為求職者王竣生本身IT產業程式領域工作經歷豐富，多以大型企業集團ERP(Enterprise Resource Planning)與DMP(Data Management Platform)兩大類系統為工作主體，並且擔任MIS/DBA/SQA/PG一系列工作職務→這兩大類大型系統後端資料庫核心功能就會提供DW(data warehouse)領域相關功能，敝人的程式開發專業也多半以此為核心→更具體來講，就是開發ETL相關資料庫後端元件與前端負責Data Visualization相關的報表Dashboard→也就如下圖所示(文章出處: 擷取、轉換和載入 (ETL) 流程 [https://docs.microsoft.com/zh-tw/azure/architecture/data-guide/relational-data/etl ])

一張含有 螢幕擷取畫面 的圖片

自動產生的描述

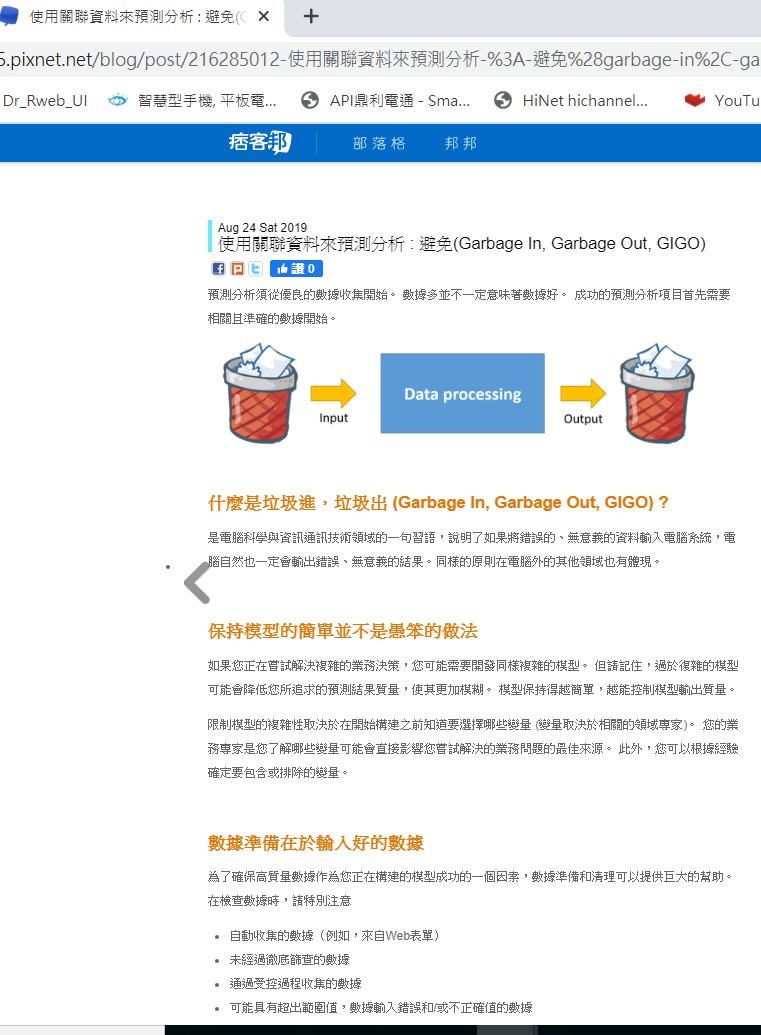
在GitHub repository(Demo2020507)所記載的專案，本身內容就是一個國際跨國集團所屬各國家分公司的特定期間總體營運→ETL後端資料處理元件TSQL Stored Procedure，就是將記載集團營運績效數字的兩張資料表，使用TSQL 關聯運算去產生集團管理階層所要的結果→然後當使用者按下Excel VBA分析模組的特定功能鍵，如下圖所示出現，用樞紐分析與圓餅長條圖表去呈現出每一個國家所屬分公司的具體銷售情況:



至於關於GitHub repository(Demo2020507)所記載的專案相關系統設計細節，可以參考以下路徑「,,,\Portfolio\TheWholePortfolioSystemDesginDocument」的以下檔案「SQL技術ETL成品用VBA程式來呈現.pdf」！而且為了驗證這一個ETL資料庫元件的成品可行度，因此在「SQL技術ETL成品用VBA程式來呈現.pdf」這一篇的第36~53頁，用初等會計學去針對ETL元件加工成品進行POC(Proof of Concept)驗證，即證明出ETL加工成品是正確無誤的！

隨後會於本GitHub repository的以下兩個目錄『NoCodeDevelopmentProduceSSRSReport』、「MicrosoftSSASBase\_MDXQuery\_Demonstration」，進行Microsoft SQL Server環境關於SSRS與SSAS相關演練與DMX分析環境架設→其中『NoCodeDevelopmentProduceSSRSReport』是將前述VBA 仿RPA的自動化會計資料檢核模組轉化到SQL Server內建SharePoint Server，因此Client 端使用者可以不用裝Excel用網頁瀏覽器就可以看到最新內容更豐富的成品→至於「MicrosoftSSASBase\_MDXQuery\_Demonstration」這一個目錄，是為了後續RPA自動化的Dashboard所需後端資料庫SSAS模組，還有MDX Cube等一系列後續開發的先遣者!

為何這一份個人作品，會花費大量篇幅去描述進行所謂資料庫層面的POC(Proof of Concept)驗證→並非只是單純展現作品作者本身MIS與DBA的專業能力，而是因為作品作者平生所經歷大型專案失敗教訓，所累積經驗讓作者自然而然有感而發的!這些經驗總結起來，就可以用一句話來總結:「Garbage In, Garbage Out」，也就是如下圖所示:



文章出處: [使用關聯資料來預測分析 : 避免(Garbage In, Garbage Out, GIGO) ]: <https://wayne265265.pixnet.net/blog/post/216285012-%E4%BD%BF%E7%94%A8%E9%97%9C%E8%81%AF%E8%B3%87%E6%96%99%E4%BE%86%E9%A0%90%E6%B8%AC%E5%88%86%E6%9E%90-%3A-%E9%81%BF%E5%85%8D%28garbage-in%2C-garba>

總而言之，要做一個有分析與預測模組的ERP或DMP大金額的大型方案，必須先有一套可用真實營業資料庫，才能讓整個系統開發流程可以由簡而難循序漸進→否則由本作品作者實戰經驗看來，這類大型專案只是在浪費國家與所屬業主單位的財政資源而已!