"Playing in tune":A mini-app for an instrument store owner

Akshay Jayakumar(230496774)
MTH 785: Programming for Business Analytics
School of Mathematical Sciences
Queen Mary, University of London

January 23, 2025

Contents

1   Introduction

The business conceived here is a musical instrument store with outlets in three cities in the UK. These stores house a variety of products; instruments in numerous models and from various parts of the world. Through the sheer variety on offer, there is something for every music lover and creator here. On display and sale are not only your run-of-the-mill acoustic guitars and violas, but a diverse collection of pieces that would leave an ethnomusicologist in raptures. While walking through any one of the stores in London, Birmingham or Manchester, one may find a South Asian harmonium sitting right next to a pair of Bongos straight from Havana. This collection is not, however, just for the "traditional" instrumentalists. Arrangers and electronic music producers may delight in the array of synthesizers, keyboards, drum machines, samplers and music production controllers(MPCs) available. The primary function of this mini-application is for myself, as the fictional store owner, to be able to update product information as new ones come in, as well as keep track of the updated database. In addition to this, the application may help in determining the level of sales (in terms of margins) I would require in order to meet certain targets. Lastly, I can survey key performance indicators in brief through the PivotTable.

2   Access functionality

The Access backend consists of four tables and three queries on these tables. The tables are titled Customers, Orders, Products and Sales. Customers contains basic information pertinent to each customer that made a purchase at a store. These may be 'member' customers that are participants of the store loyalty program, or ordinary, 'normal' ones. Each record also indicates the city in which a particular customer made a purchase. The Orders table contains information with regard to payment method used and the date an order was made. As I attempted to change headings in this table, the table records vanished. However, this has not affected query implementation or data loading, and the Orders table is shown below for reference(see table below). Products outlines each product held by the stores, the category in which the product falls(electronic/non-electronic), and ratings which are aggregated through the reviews willing customers leave us. A key point to note here is that certain instruments come in a variety of models, and so these have been assigned unique product IDs in the Products table. Finally, the Sales table, as the name may indicate, contains proprietary information as regards individual unit prices and costs, the quantity ordered(customers don't usually just buy single pieces), revenue generated and total costs, and the income and margin represented in percentage value. Moving onto the SQL queries, the first one displays the total income generated by each category of instrument, which helps me gauge the type of demand for my products, and to whom I must direct my marketing efforts towards. After all, the music historians cannot be sold to in the same way one might electronic music producers. The next query outputs the number of samplers/MPCs (this is a single product in the Products table) sold in each store location. These are priced somewhere between the non-electronic instruments, and the pricier synthesizers, and so I thought it worthwhile to see how they were being sold in each location. This can serve as an introductory point for further analysis, to see how they may be priced and which models I may acquire to be sold

in the future. The last query shows the number of instruments purchased/ordered in each month of 2019(the year in question being examined, for which data is available). This sort of query helps monitor possible seasonality in purchase habits, which consequently enables me to adjust my prices and marketing methods accordingly.

Invoice ID	Customer ID	Payment	Order date	Ship date		
750-67-8428	iza5ft	E-wallet	01/05/2019	01/07/2019		
226-31-3081	07s0ls	Cash	03/08/2019	03/19/2019		
631-41-3108	ke5eh6	Credit card	03/03/2019	03/10/2019		
123-19-1176	gltii3	E-wallet	01/27/2019	01/30/2019		
373-73-7910	yj9u0f	E-wallet	02/08/2019	02/18/2019		
699-14-3026	g9gjhe	E-wallet	03/25/2019	03/29/2019		
355-53-5943	993baf	E-wallet	02/25/2019	02/26/2019		
315-22-5665	rjaaay	E-wallet	02/24/2019	02/28/2019		
665-32-9167	cs19gz	Credit card	01/10/2019	01/15/2019		
692-92-5582	0yxqf0	Credit card	02/20/2019	02/22/2019		
351-62-0822	ke5eh6	E-wallet	02/06/2019	02/10/2019		
529-56-3974	o0lxc0	Cash	03/09/2019	03/13/2019		
365-64-0515	te0cqq	E-wallet	02/12/2019	02/14/2019		
252-56-2699	lu8g2g	E-wallet	02/07/2019	02/11/2019		
829-34-3910	3333hi	Cash	03/29/2019	03/30/2019		
299-46-1805	993baf	Cash	01/15/2019	01/19/2019		
656-95-9349	gltii3	Credit card	03/11/2019	03/16/2019		
765-26-6951	pykco5	Credit card	01/01/2019	01/04/2019		
329-62-1586	cs19gz	Credit card	01/21/2019	01/23/2019		
319-50-3348	np19lk	E-wallet	03/11/2019	03/17/2019		
300-71-4605	yj9u0f	E-wallet	02/25/2019	02/27/2019		
371-85-5789	rjaaay	E-wallet	03/05/2019	03/08/2019		
273-16-6619	te0cqq	Credit card	03/15/2019	03/20/2019		
636-48-8204	ke5eh6	E-wallet	02/17/2019	02/21/2019		

Fig. 1: The Orders table

3 Excel features

Now it bears examining what the app actually does, or rather the functionality of the frontend. The "app" is split into three sheets, each of which performs a particular operation, or set of operations. The first one interacts with the Products table in the aforementioned Access database, allowing the user to add new records, as well as see any updates made to the dataset. Both functions can be carried out with the literal click of the respective button, placed emphatically next to the load and input areas. The next table contains preloaded data from the Sales table in the database, and has a "Target GMP" labeled cell, which contains a pre-defined formula. Using Solver, I aimed to see how prices can be set to meet different margin percentage expectations. This is done through the use of Solver which takes a range, an input target value and any constraints and modifies the range using a solving method. My attempts at recording a macro and creating a point-and-click or otherwise "automated" method to do this failed, and I admit I do not fully understand what the issue was. The final sheet combines the Products and Sales tables, whose data was loaded through a connection established with the Access database, to form a PivotTable that displays key values such as cumulative gross income and average ratings and GMP(gross margin percentage) across all available products. Some values are missing as certain products were entered using the data

entry system, however, the Sales table needs to be updated for these items, and can easily be done so.

4 VBA broad-strokes functionality

VBA was largely used in the creation of the first bit of functionality; namely, data entry and loading. There are three subroutines in total for these tasks. The first one, LoadFromDatabase(), "loads" product-related data from the Products table in the Access database. It does this by defining a connection to the database, providing a SQL query to select all records in the table and place it in a Recordset object, adding column headings using a For loop, and copying the data from the Recordset object onto the sheet. The second subroutine, SaveToDatabase(), while as straightforward in function, in that it takes data input by the user and saves it to the database, is not quite so in terms of implementation. Broadly speaking, this requires defining an input area, using a For loop to read data, using data validation to ensure that only valid entries are accepted, and SQL instructions at the end to insert this data into the Products table in the database. We may use the first query, made accessible through a point-and-click button, to check and see whether our values have been input, even as there is a message box output command at the very end that seeks to confirm this. The third subroutine is a repetition of the first, except that it loads the Sales data from the sheet in the database to be worked on using Solver.

5 Conclusion

There is a sense of real wanting while reviewing this project, not so much from a conceptual standpoint as much as that of a technical one. I am no 'code sculptor'; my approach to programming in any language is largely dependent on using existing blocks in potentially new ways. I do not quite care to remember every bit of syntax, though I understand structure and semantics, or the logic of what is being done in each line. I am fond of finding redundancies and eliminating them. That being said, it is regrettable that I have had to resort to repetitive VBA code in this project. It was not my first solution, and I truly consider this a 'work in progress', grade contingencies notwithstanding. There are issues within the features of the code, such as having to hard-code a file path in each subroutine for them to work. I was not able to find a workaround for this; I am not familiar with concepts such as relative paths. I dislike the use of large-language models(so-called "AI") for writing or anything else at all, but due to my aforementioned approach I do see merit in using them for programming. I must admit to having used them for this project, particularly in creating the VBA queries, alongside consulting the class material. I used an existing supermarket dataset found on Kaggle as a starting point for this project, though it bears little to no resemblance post-modifications. As for the potential in this project, I see a possible expansion in product range, particularly in the inclusion of physical media such as vinyl records, which has a great deal of customer overlap with the existing customer base. Furthermore, events could be included as an offering, where various local musicians and DJs are hosted periodically at the stores. These avenues offer new dimensions of potential insight, and is an

exciting prospect for additional functions within the app. This app could "moonlight" as an event management assistant that could help artists as well as store owners. I can grasp the potential, but the technical side is a work in progress, and I hope it is assessed as such.