

Student Number  
180055393

Database Application Report

## Table of Contents

1. How to start and work the programme .....	2
2. Most significant features .....	4
3. In-depth Functionality View / Non-Trivial Queries .....	8
4. Management Reports and their purpose.....	11
5. Non-Trivial Charts .....	13
6. Testing the Database and making new Data .....	15
7. Reflection .....	20

## 1. How to start and work the programme

---

In order to open the database, locate the file named “**Database\_Assingment\_Individual**” in your computer using the search bar as shown below and clicking on the named file.

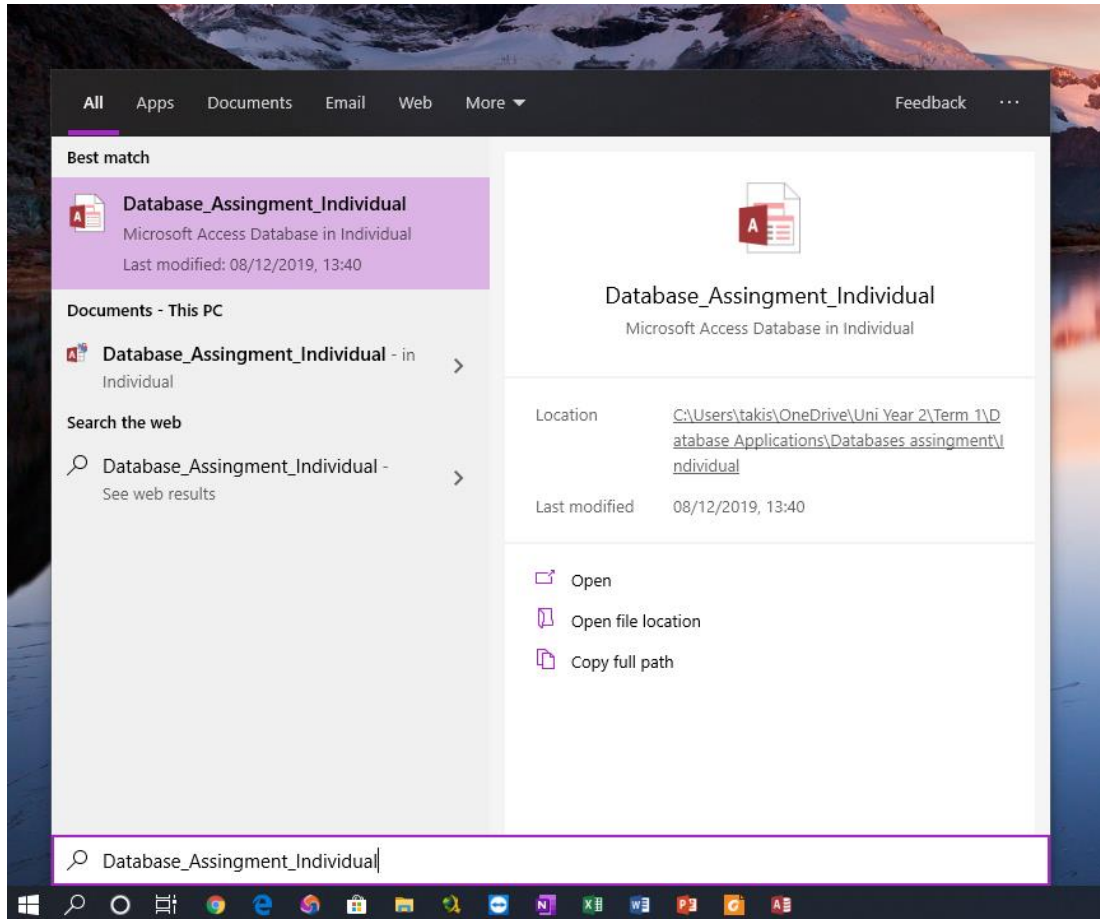


Figure 1 : Locate File

When you click on the file the follow screen will appear. You are now in the programme and you are looking at the **Home Page**. To get into the **Menu page** with all the functionalities, click **Menu**.

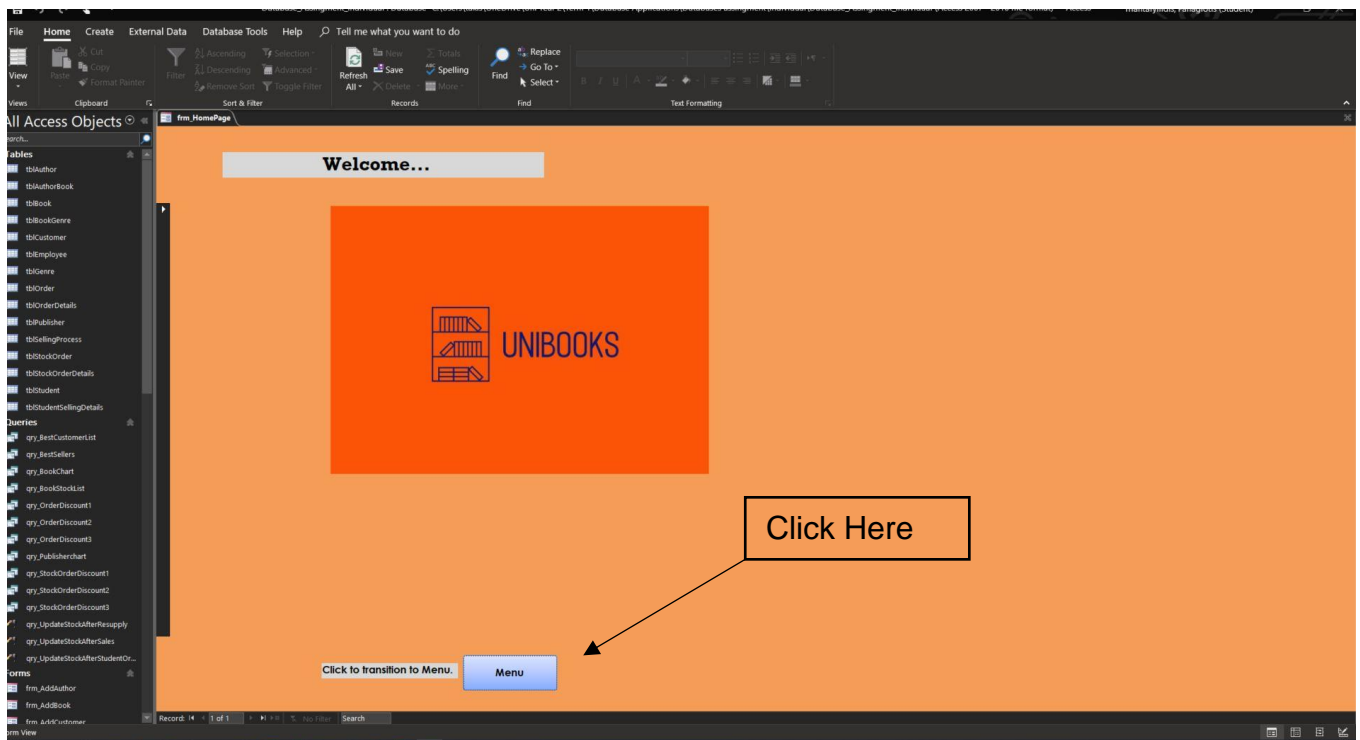


Figure 3 Home Page

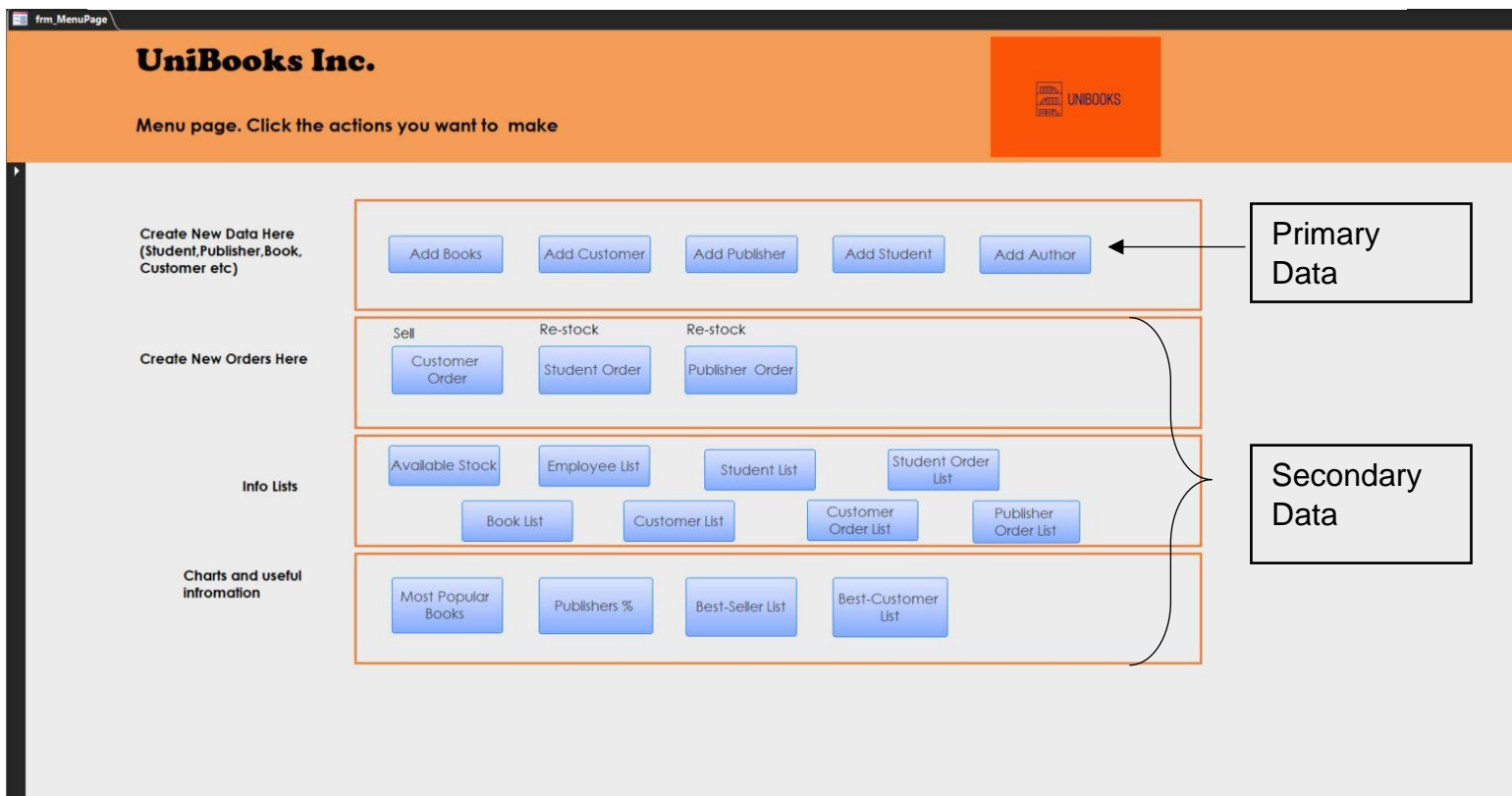


Figure 2: Menu Page

Inside the Menu is a number of many different useful actions that will help in the process of operating your business. Information such as Publishers, Students, Customers, Books and Authors can be characterised as Primary Data and functions such as Orders and List can be named Secondary Data, since they require existing Primary Data to work. Each type of actions has its own category box; e.g. The necessary buttons for creating new primary data are all located inside the “Create New Data Here” box. Each button, box and page has been carefully named to match their functionality so pay attention to the labels and names to guide in the programme.

## 2. Most significant features

The first most useful feature of the application is the ability to create all the necessary Primary Data required to compose records of orders, charts, Info Lists etc. which will be useful in managing demand and supply of the business. The user is able to register new Customers, Publishers, Authors and Student sellers at any given moment, with the flexibility to navigate across different “forms” without losing any data. As shown in the figures below.

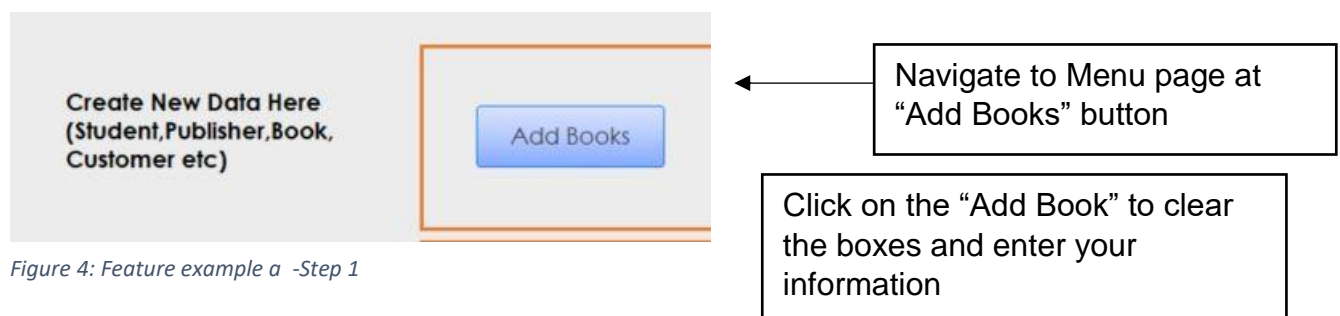


Figure 4: Feature example a -Step 1

**Add a New Book**

ID

Book Title

Cover Price

Publisher

Edition

Language

Year

Stock

Save

Add Book

Delete Book

Click HERE Assign Author

To Add an Author AFTER you create the Book,

<- Can't find your Publisher? Click HERE New Publisher

Get back to Menu with saved progress.

Menu

Fully Exit the process

Exit

Figure 5: Feature example a -Step 2

After you enter the information click "Save". If you can't find your Publisher follow the helpful labels as shown in the image. When you are finished, assign an author by clicking the button and transitioning to the appropriate window.

## Add a New Book

ID

Book Title

Cover Price

Publisher

Edition

Language

Year

Stock

Save

Add Book

Delete Book

<- Can't find your Publisher? Click HERE

Get back to Menu with saved progress.

Fully Exit the process

To Add an Author AFTER you create the Book,  
Click HERE

Figure 6: Feature example a - Step 3

## Alocate a book to its Author

Here

ID

Book

Author

Save

Add

Delete

Can't find your Author? Click HERE

Assign each book with their author. In the case of missing data follow the same process as above. **Buttons have same functions based on colour.** Don't forget to "Save".

Figure 7: Feature example a - Step 4

Another useful feature of the programme is Visibility and helpful UI (User Interference). Not only it aims to provide all information organised for the user to view, but also it follows a simple and minimalistic design. Helpful labels, button names, Titles and subtitles, lists etc. all provide only the most useful information to help users navigate through the application without getting lost and confused. As shown above, there is flexibility to move across different function 'forms' and create data while at the same time you may view the pre-existing records to get any information as you may see from the figures below.

You may be in the process of creating a resupply order with a Publisher, clicking the **"Publisher Order"** and at the same time you wish to view the full list of the previous orders made. You can do so by clicking in the Menu button to go back to Menu, without closing the form, as indicated in the label, and click on the **"Publisher Order List"**.

**UniBooks Inc.**

Place Resupply order for a book here.

ID:

Date:

Publisher:

Employee:

Place New Order

Save and Update

Delete Order

Invoice

Details

StockOrderDetailsID	StockOrderDetails_Qty	Book
1	20	Law for Business
(New)	0	

Record: 1 of 1

Get back to Menu with saved progress.

Fully Exit the process.

Menu

Exit

Figure 8: Feature example b - step 1

As shown above, there are two ways to exit. Either go back to Menu and open a new window or fully Exit this one.

Now you can view the list with all the previous registered resupply orders. As the arrow indicates, the previous window is still open where it was left. That means, by going back and clicking the “Publisher Order” again, you will still be able to work where you left while browsing the other windows by either clicking on them in the top bar or pressing their button.

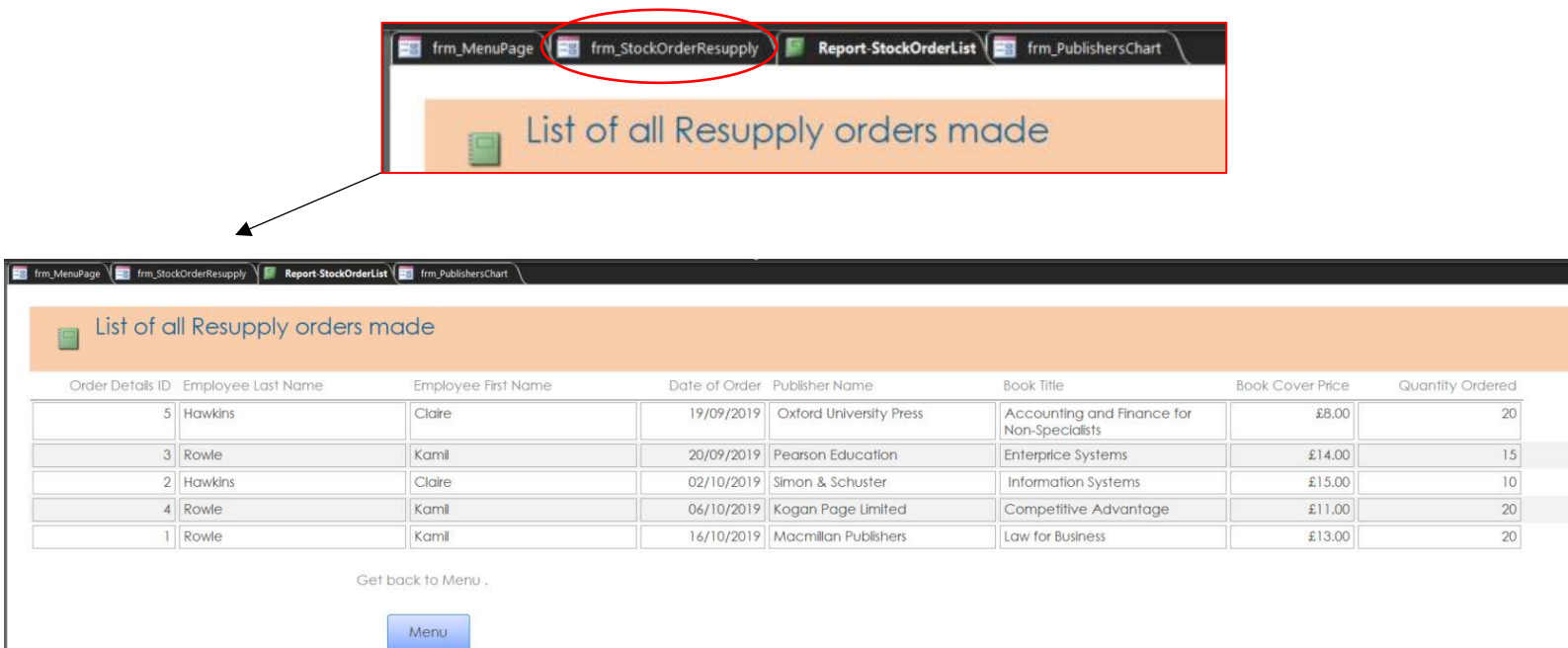


Figure 9 : Feature example b - step 2

The programme will organise the information that is made into list; 8 lists all providing the information shown clearly by the names. Simply click on one and you can access all registered data of that information, same way as shown by the example above.

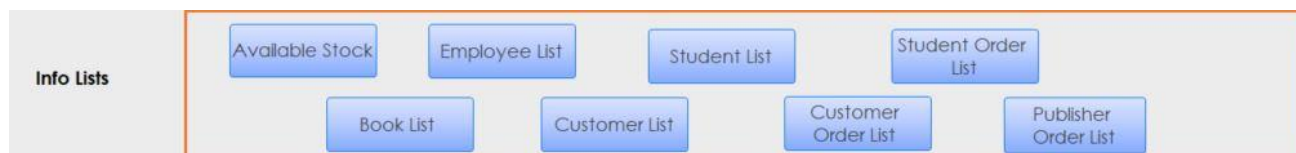


Figure 10 : Feature example b - step 3



### 3. In-depth Functionality View / Non-Trivial Queries

A query is a request of information from one or more tables in the database linked together by the ER connections, in order to show an output that can be viewed by the user. Queries can perform calculations, establish attributes and set criteria which will affect the output. Some examples might be the creation of Info lists, reports, Charts etc. The queries shown below are called Non-Trivial / type 2 because they require information from more than one tables to function, and they make calculations. Here is an example.

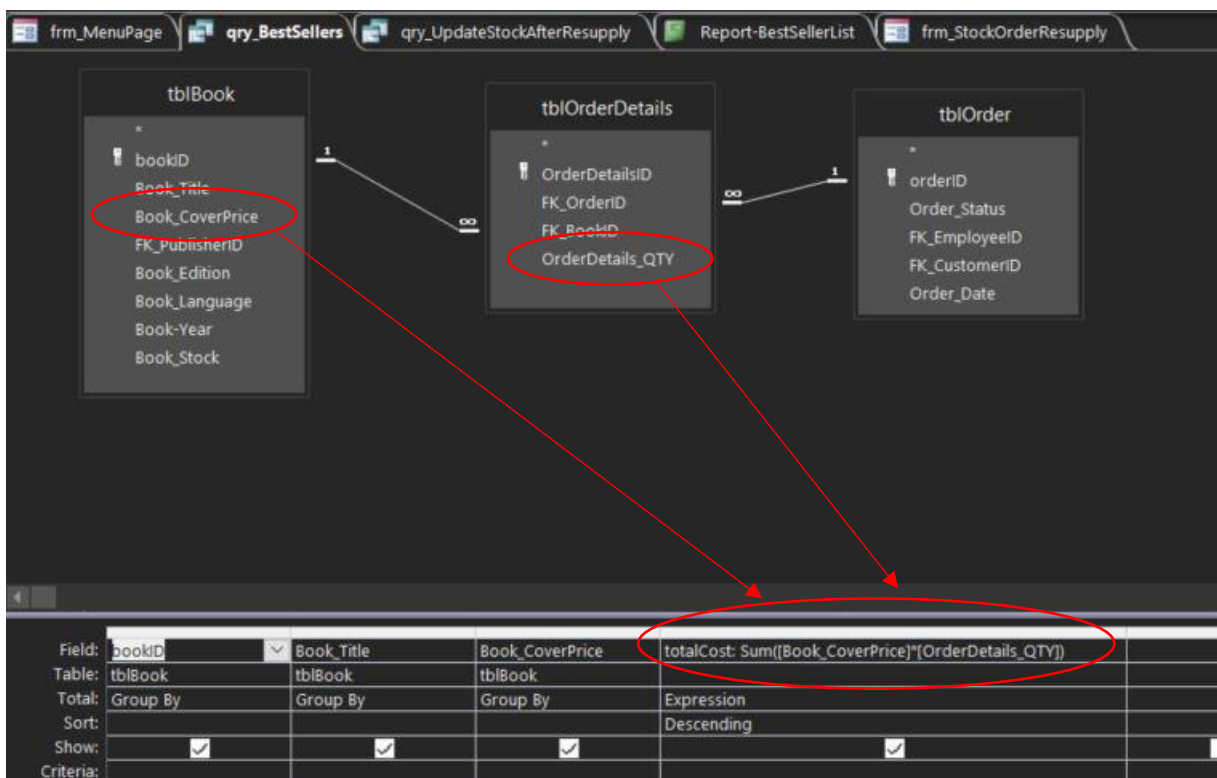


Figure 11: First Non-Trivial Query\_a

The Above query's purpose is to get information from 3 different tables and create a list with all the Books and their total revenue generated, in a descending order, providing the "Best-Seller" books at the top. As shown by the red markers, two attributes from two connected tables are combined into calculating a newly created attributed named "TotalCost" which will be used to show the total generated revenue for each book.

Below is an example of the output of this query.

Best Sellers			
bookID	Book Title	Cover Price	totalCost
2	Information Systems	£15.00	£675.00
5	Competitive Advantage	£11.00	£330.00
12	Law for Business	£13.00	£169.00
6	Accounting and Finance for Non-Specialists	£8.00	£144.00
4	Enterprise Systems	£14.00	£140.00
13	Lawless Business	£7.00	£98.00
1	Analysis Tools	£12.00	£36.00

Get back to Menu .

Menu

Figure 12: First Non-Trivial Query\_b

Queries can also have more “active” functions. Besides showing pre-made information, structured in a list, they can also be used for **Updating** attribute information. The next query is used after every new order is made (selling or resupplying) and ensures the stock of a book is increased (after resupply order) or decreased (after selling order) meaning its updating the stock when we press the button. Following is an example of its functionality.

The screenshot shows a database application with several open windows: 'frm\_MenuPage', 'qry\_BestSellers', 'qry\_UpdateStockAfterResupply', 'Report-BestSellerList', and 'frm\_StockOrderResup'. The 'qry\_UpdateStockAfterResupply' window is the active one, displaying a query editor. The query is designed to update the 'Book\_Stock' field in the 'tblBook' table. The query text is: 'Update To: [Book\_Stock]+[StockOrderDetails\_Qty]'. The criteria are: 'StockOrder\_Status' is 'Yes'. The query is also linked to the 'tblBook' and 'tblStockOrder' tables, which are shown in the background. The 'tblBook' table has fields: bookID, Book\_Title, Book\_CoverPrice, FK\_PublisherID, Book\_Edition, Book\_Language, Book\_Year, and Book\_Stock. The 'tblStockOrder' table has fields: stockorderID, StockOrder\_Status, StockOrder\_Date, FK\_PublisherID, and FK\_EmpolyeelD. The 'tblStockOrderDetails' table has fields: StockOrderDetailsID, FK\_StockOrderID, StockOrderDetails\_Qty, and FK\_BookID. The query is also linked to the 'tblStockOrderDetails' table, which is shown in the background. The 'tblStockOrderDetails' table has fields: StockOrderDetailsID, FK\_StockOrderID, StockOrderDetails\_Qty, and FK\_BookID. The query is also linked to the 'tblBook' table, which is shown in the background. The 'tblBook' table has fields: bookID, Book\_Title, Book\_CoverPrice, FK\_PublisherID, Book\_Edition, Book\_Language, Book\_Year, and Book\_Stock. The query is also linked to the 'tblStockOrder' table, which is shown in the background. The 'tblStockOrder' table has fields: stockorderID, StockOrder\_Status, StockOrder\_Date, FK\_PublisherID, and FK\_EmpolyeelD. The query is also linked to the 'tblStockOrderDetails' table, which is shown in the background. The 'tblStockOrderDetails' table has fields: StockOrderDetailsID, FK\_StockOrderID, StockOrderDetails\_Qty, and FK\_BookID.

Figure 13: Second Non-Trivial Query\_a

It also updates the status check-box of an order, making it “Yes” if the order was not processed and now it is. Making sure we keep track of which orders we’ve made and which not.

As we can see, it adds-up the quantity of a book ordered for resupply, to the overall Stock of that book, changing the number in the database and in our reports/lists. This query is inside the “Save and Update” button, located in every Order creation form. The user only has to press this button once every time they make an order.



Figure 14: Second Non-Trivial Query\_b

Let's analyse the SQL code behind this query for a greater understanding on its functions.

```
UPDATE tblStockOrder INNER JOIN (tblBook INNER JOIN tblStockOrderDetails ON
tblBook.bookID = tblStockOrderDetails.FK_BookID) ON tblStockOrder.stockorderID =
tblStockOrderDetails.FK_StockOrderID SET tblBook.Book_Stock =
[Book_Stock]+[StockOrderDetails_Qty], tblStockOrder.StockOrder_Status = Yes
WHERE (((tblStockOrder.StockOrder_Status)=No));
```

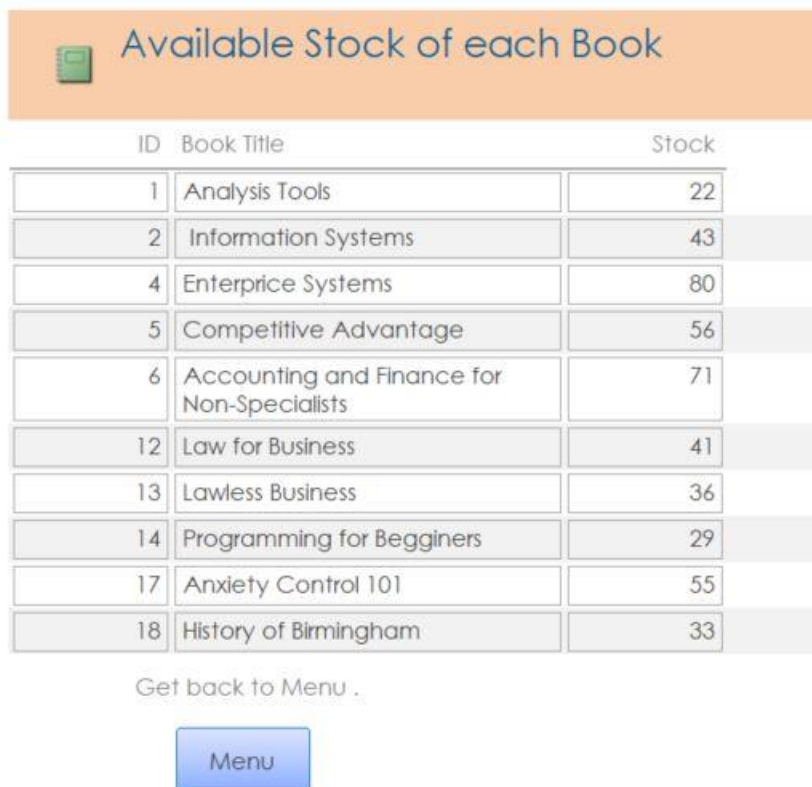
The code first indicates that this is an Update process, meaning the information of an attribute will be replaced by the following process. Then allocates where each attribute originates from, setting the StockOrderDetails\_QTY (the quantity ordered) inside the BookStock column in order to perform the calculation. Then, with a “SET” statement it changes the value of the attribute to the following calculation included in the brackets [ ].

The comma (,) suggest that the Update process still continues but for a second “act”. It states that it will update to “Yes” everywhere that status attribute is set to “No”. Meaning it will check the boxes of the orders which just got processed.

## 4. Management Reports and their purpose

---

The purpose of reports in the programme is to provide us with useful information, well-structured in the same page and in the simplest way possible. This information can vary from statistics to chart and Lists. They gather their information from queries and forms where we structure the connection of entities and create new data, which will then be displayed in the report. An example of a very useful report is the “Available Stock” report.



ID	Book Title	Stock
1	Analysis Tools	22
2	Information Systems	43
4	Enterprise Systems	80
5	Competitive Advantage	56
6	Accounting and Finance for Non-Specialists	71
12	Law for Business	41
13	Lawless Business	36
14	Programming for Beginners	29
17	Anxiety Control 101	55
18	History of Birmingham	33

[Get back to Menu .](#)

[Menu](#)

Figure 15 : Report 1

As we can see, this report provides us with a simple, yet important, list of all the currently registered books, their IDs and their available stock. This means an employee can view this information at any time, and if they judge appropriate or if a limit has been reached then they are able to respond in time and request a new resupply order from a Publisher.

Another useful type of report is the Invoice. An invoice is generated with the pressing of the button included in every Order-making form and it provides the user with all the important information taken from the newly created order. This is very important for the company to be able to keep track of its activities and provide this information to buyers and sellers whenever needed.

# Order Invoice

a

Customer First Name:  Date:

Customer Last Name:  orderID:

Book Title	Cover Price	Quantity Sold	Cost
<input type="text" value="Analysis Tools"/>	<input type="text" value="£12.00"/>	<input type="text" value="3"/>	<input type="text" value="£36.00"/>

b

Discount (%) :

Cost:  
(With discount)

Total:  
(with packaging fees) +2.95

c

Return to the previous page

Figure 16: Example of Invoice (from Customer Order form)

The Invoice a) will show the individual's and the product's information along with unique order ID and date of transaction. B) It will then calculate a possible discount based on the company's algorithms. C) It will use the discount to calculate the final price of the product towards/from the buyer/seller.

## 5. Non-Trivial Charts

---

A chart is called Non-trivial because it requires more than one tables to gain enough information to function as its meant to. The following chart provides useful data about which books are more popular among the customer, showing the ones with total quantity sold from all the orders to be more than 20..

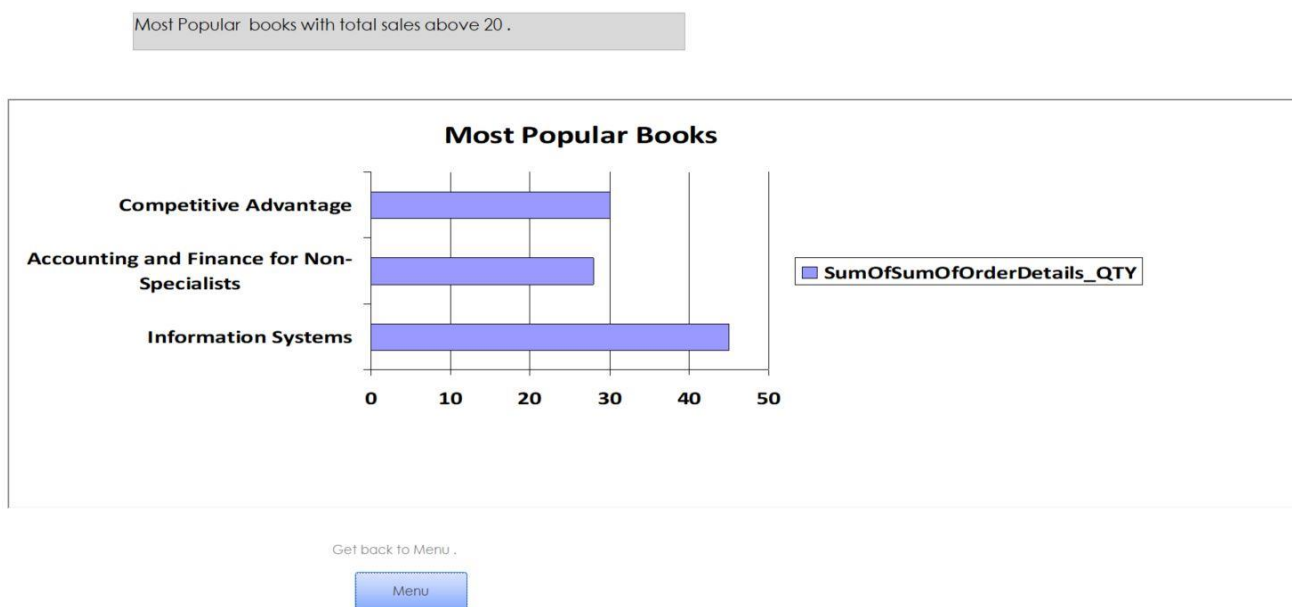


Figure 17: chart 1

We can clearly see three books are the most famous amongst the rest totalling the numbers as viewed above.

Below is the query behind this chart.

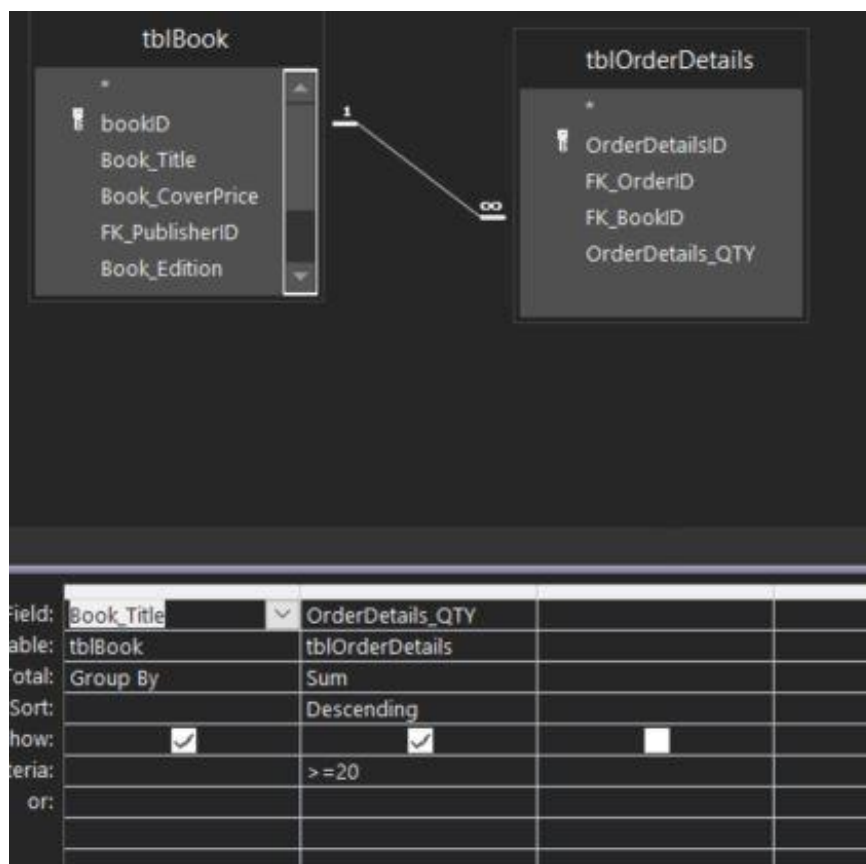


Figure 18: Chart query

In a few simple statements, this query groups all the books by their name to avoid getting repeated and sums up all the quantities sold. Then the criteria ensure that only the books with total quantity sold above 20 will be shown.

## 6. Testing the Database and making new Data

---

Throughout the database, various criteria and validation rules have been set to ensure the database is error free up to a certain extend.

Example of a validation rule is when we assign a box for numbers only, in case the user enters text.

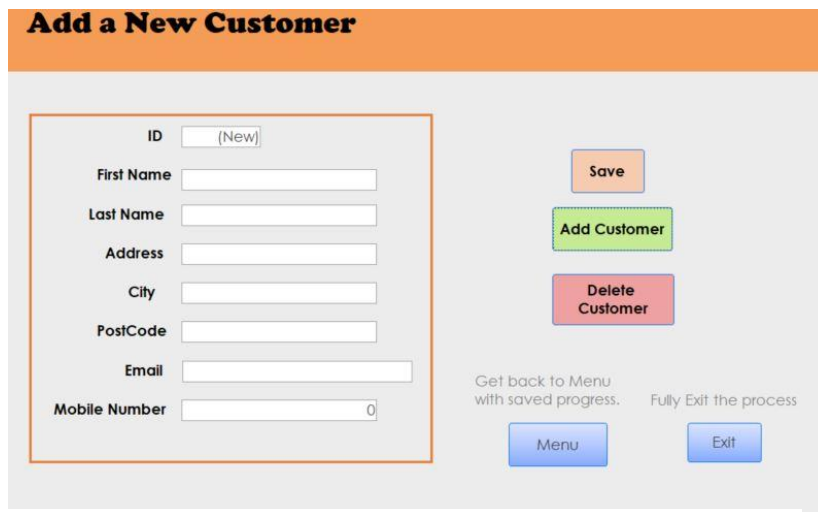
The screenshot displays a Microsoft Access form with several input fields: ID (containing '7'), First Name, Last Name, Address Line1, Address Line2, Post Code, Mobile Number (containing 'text example for error'), and Email. A blue 'Save' button is positioned above a red 'Delete Student' button. A Microsoft Access error dialog box is open, stating: 'The value you entered isn't valid for this field. For example, you may have entered text in a numeric field or a number that is larger than the FieldSize setting permits.' Below the dialog, there are two options: 'Get back to Menu with saved progress.' and 'Fully Exit the process.' At the bottom, there are 'Menu' and 'Exit' buttons.

Figure 19: Error message example

In order to ensure the database is working properly, the following process will show step by step the creation of a new Customer, a new Employee, a new Customer order including the new Primary Data, a generated invoice and the orders being included in the overall Order List.



Step 1: Open Menu, click on “New Customer”, click “Add Customer” to clear the boxes



The screenshot shows a web form titled "Add a New Customer" with an orange header. On the left, a light gray box contains several input fields: ID (with a "(New)" placeholder), First Name, Last Name, Address, City, PostCode, Email, and Mobile Number (with a "0" placeholder). To the right of these fields are three buttons: "Save" (orange), "Add Customer" (green), and "Delete Customer" (red). Below these buttons are two blue buttons: "Menu" and "Exit". Text between the buttons reads "Get back to Menu with saved progress." and "Fully Exit the process".

Figure 20: Step 1 \_ Create New Customer

Step 2: After entering the information, click “Save”, then go back to menu and press “Add Employee”.



This screenshot shows the same "Add a New Customer" form, but now with sample data entered into all the input fields. The data is: ID: 12, First Name: Nikos, Last Name: Triantafyllidis, Address: 22, New Street, City: Birmingham, PostCode: D5 8ER, Email: NikTrian@gmail.com, and Mobile Number: 7595833600. The "Save" button is now highlighted with a dashed border, indicating it is the next step in the process. The "Add Customer" and "Delete Customer" buttons remain visible below it, along with the "Menu" and "Exit" buttons at the bottom.

Figure 21: Step 2\_enter information and click Save

## Add a New Employee

ID

5

First Name

George

Last Name

Baxter

Date of Birth

22/07/2000

Address

30, New Street

City

Birmingham

PostCode

G6 7PM

Email

GeoBaxter@hotmail.com

Department

Sales and Distribution

Employment Date

14/06/2018

Mobile Number

7558534599

Save

Add Employee

Delete Employee

Get back to Menu with saved progress.

Fully Exit the process

Menu

Exit

Step 4: Click “Customer Order” and again click “Add New Order to empty the boxes. Then enter your Information you just made.

Figure 23: SStep 4\_ open New Order form\_ clear the boxes and enter your new information

When you enter the information, you may want to “Save and Update” to update the new stock. The following message will appear .

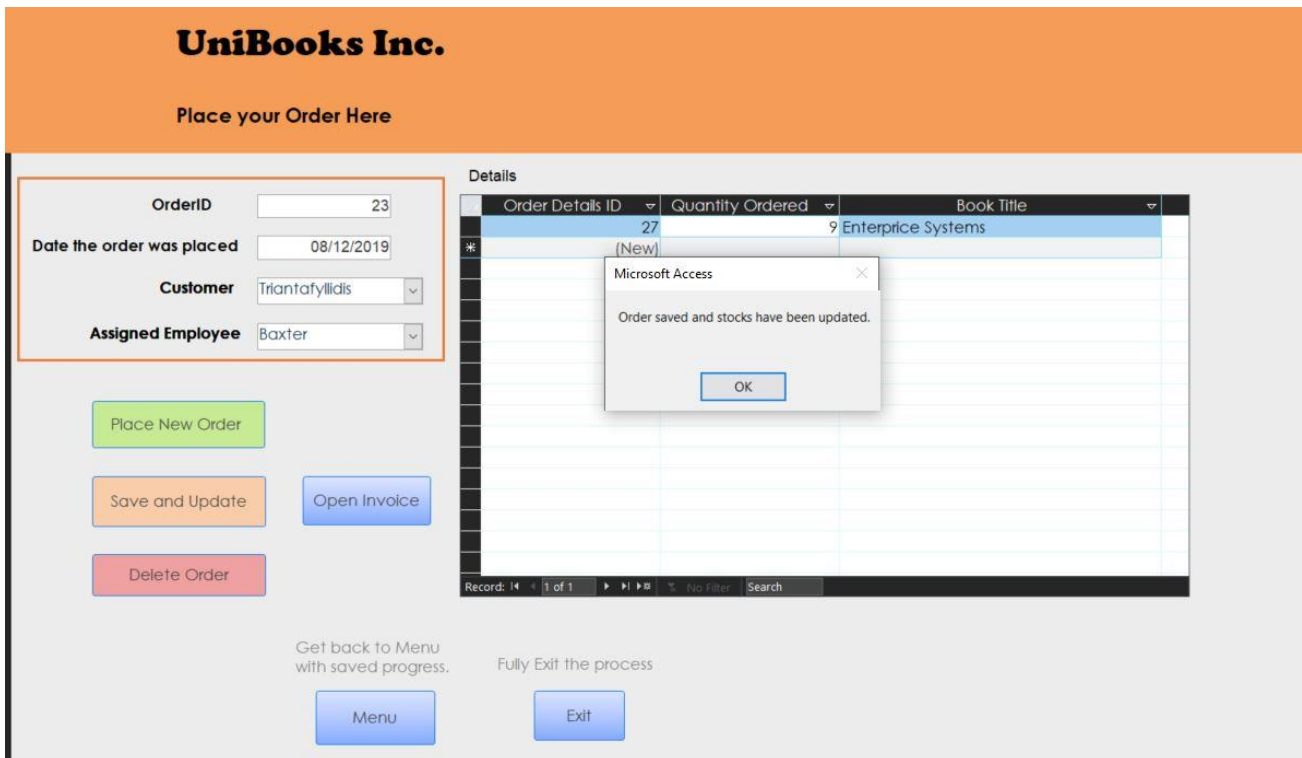


Figure 24: confirmation message

Finally open the Invoice and confirm all the information is correct. If yes, then you have successfully finished the demo functionality test.

## Order Invoice

Customer First Name	<input type="text" value="Nikos"/>	Date	<input type="text" value="08/12/2019"/>
Customer Last Name	<input type="text" value="Triantafyllidis"/>	orderID	<input type="text" value="23"/>

Book Title	Cover Price	Quantity Sold	Cost
<input type="text" value="Enterprice Systems"/>	<input type="text" value="£14.00"/>	<input type="text" value="9"/>	<input type="text" value="£126.00"/>

Discount (%) :

Cost:  
(With discount)

Total:  
(with packaging fees) +2.95

Return to the previous page

## 7. Reflection

---

Throughout the development of the programme I personally found it challenging to have in mind all the various connections between data that effects functionality in different aspects of the database. Making a mistake is critical as it creates a long set-back, trying to find something small out of place, in a big database. It was educational and exciting nevertheless to see and understand how a database work.

There were still some limitations. Here are a few of them.

- There is no search engine to type words and show relevant information.
- In the Publisher Order, the list of book options is not limited to the selected Publisher.
- Author cannot be viewed from any lists.
- Discount is applied to every Publisher order instead of being optional

And here are some of the strengths of the application:

- Clear, simple and easy to use UI
- Documentation Lists storing every important information (orders, Primary Data)
- Useful Charts and Lists (Best -Sellers, Best Customers, Publisher percentages of our total book stock in the business)

Ways of improving the Database:

- Adding a search bar in every list for finding data faster.
- More charts showing the Employee performance.
- Invoice-saving mechanism
- Automatic resupply process based on re-order limits set for each book