

KG1902428 Database Design (The Book Store) (ACCB4013)

by Kallum Gordon

FILE	17858_KALLUM_GORDON_KG1902428_DATABASE_DESIGN__THE_BOO K_STORE__ACCB4013__252092_1530943045.DOCX (184.26K)		
TIME SUBMITTED	06-MAY-2020 03:57AM (UTC+0100)	WORD COUNT	2048
SUBMISSION ID	126383051	CHARACTER COUNT	9736

Database Design (The Book Store)

Author: Kallum Gordon

Prepared for: B Holland

15th April 2020

Database Design (The Book Store)

Table of Contents

1.0 Abstract	3
2.0 Systems specifications	3
2.1 Entity relationship diagram (ERD)	3
2.2 Tables.....	4
2.2.1 Books	4
2.2.2 Customers	4
2.2.3 Locations.....	4
2.2.4 Orders.....	4
2.2.5 Shop	4
2.2.6 Staff.....	4
2.2.7 Warehouse.....	5
3.0 Database solution.....	5
3.1 Data types and Input masks	5
3.1.1 Books table.....	5
3.1.2 Customers table.....	5
3.1.3 Locations table	5
3.1.4 Orders table	5
3.1.5 Shop table.....	6
3.1.6 Staff table	6
3.1.7 Warehouse table.....	6
3.2 SQL.....	7
3.2.1 Coding a table, relationships, and adding information within SQL.....	7
3.2.2 Coding queries within SQL.....	8
3.3 Additional queries	9
3.3.1 Producing a report from a query	9
3.4 Main Menu.....	9
4.0 Conclusion	10
5.0 Appendices.....	10
Appendix A: SQL Code.....	10
Orders query:	10
All book titles query:	10
Search book ID query:	10
Insert values into Orders table:	11

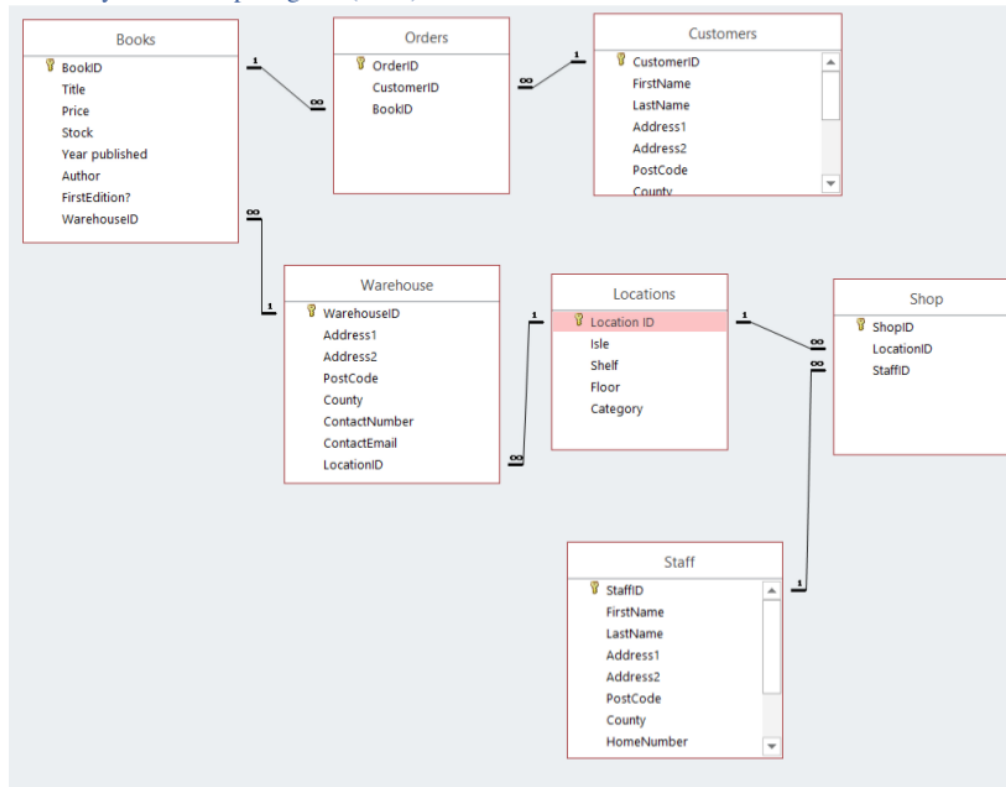
Database Design (The Book Store)

1.0 Abstract

Within this report, documentation has been produced to support a synthesised solution to a book store database, the database has also been created with all requested features.

2.0 Systems specifications

2.1 Entity relationship diagram (ERD)



An ERD is designed to allow a Database administrator to plan out their Database layout in advance. This can be helpful to visualise the structure of the Database before committing to creating it, this can save countless days of time when coding the Database through SQL. An ERD is essential for having good Database design.

The books table is the main Database in this ERD as it can link to every other table. There can be one book to many orders, meaning a single book may be ordered many times. There can be many orders from one customer, therefore within the orders table, customer ID links to the customers table. Each relationship stemming from the orders table was coded by hand within SQL. Many books can be stored in one warehouse, therefore within the books table, warehouseID has a one to many relationship to the Warehouse table. There can be many book locations within one warehouse, there can be one location for a book which can be sent to many book shops. Many workers can work in multiple shops, depending on their rota. This then produces appropriate master and embedded forms within the tables.

2.2 Tables

2.2.1 Books

The main table in the Database. This is the main focus as customers come to the shop for them. It stores book titles, the price, the stock, the year published, the author, whether the book is a first edition or not as the book store has a focus on this, and the ID of the warehouse the book is stored in.

2.2.2 Customers

This table is critical as it stores the details of customers who have created an account, the business relies on this table to run. Stored within it is the customers unique ID, the customers first name, their last name, their first address (house number and street name), their second address, their post code with an appropriate input mask, their county or country, their home phone number if applicable, their mobile number, their email address, their password, their salt password, whether they're subscribed to the newsletter or not, and a custom comment the customer can place on their profile for customisation and expression. Of note within this information is the use of a salt password, this hashes the data and is to store the customer's password in an encrypted form to safeguard it within storage.

2.2.3 Locations

This table documents the locations of each book within a warehouse, it notes the isle it is stored on, the shelf it is kept on, the floor it is located, and the category of book such as mystery, thriller, horror, slice of life, romance, or classified.

2.2.4 Orders

Orders keeps track of orders placed by customers. It assigns each order a unique ID, which then links to the customer ID and book ID. It links to the customer ID to easily be able to see the address of the customer who ordered to be able to swiftly mail the book. It also links to the book ID to easily tell which book has been ordered, the stock can then be directly altered to represent the new data.

2.2.5 Shop

This table was created to allow for the possibility of multiple book shop locations, each shop has a staff member or two manning it at any one time to mail books and keep the inventory updated.

2.2.6 Staff

The staff table documents the information of each staff member within the employ of the book company. Each staff member has a unique ID, additionally their first name, their last name, their first address (house number and street name), their second address, their post code with an appropriate input mask, their county or country, their home phone number if applicable, their mobile number, their email address, their password, and finally their salt password.

2.2.7 Warehouse

Ideally the book store would have many warehouses within their contact to source books when needed. They store the warehouse ID, the first address, the second address, the post code, the county it is located, its contact number, its contact email, and the location ID to find the books within.

3.0 Database solution

3.1 Data types and Input masks

Input masks are necessary as they allow the administrator of the Database to apply restrictions on data entry, for example when entering a postcode, three letters or numbers could be required, then a forced space, followed by four more letters or numbers. This saves time troubleshooting and prevents simple mistakes from occurring such as adding an extra number onto the publish date of a book.

3.1.1 Books table

The price has an input mask that forces the first character to be a pound sign, it then allows up to five additional digits. The stock allows for up to four digits. The year published forces the user to enter four digits.

3.1.2 Customers table

As The City Book Store mail to locations outside of Swansea, the postcode input map allows for 7 characters.

The county data type has been set to long text to account for very long Welsh county names.

The home number field allows for up to 11 digits. The mobile number field also allows for up to 11 digits. This disallows unrealistically long numbers. These are both set to short text instead of number to allow for 0s.

The password is set up in a way that forces the first character to be uppercase, a user can enter up to 20 characters, one of which must be uppercase. Salt then further encrypts this.

Customer profile comment allows them to post a comment below their profile on the website, as a customisation option.

3.1.3 Locations table

This table forces the user to enter two digits for the isle, two digits for the shelf, and one digit for the floor, this could be altered to two if the book company expanded.

3.1.4 Orders table

This table has no input masks as it only contains IDs for relationships.

3.1.5 Shop table

This table has no input masks as it only contains IDs for relationships.

3.1.6 Staff table

The staff table is set up exactly the same as customers table. Including the password and postcode input masks.

3.1.7 Warehouse table

This table has an input mask on the post code, it also has one on the contact number, restricting it to eleven digits.

3.2 SQL

3.2.1 Coding a table, relationships, and adding information within SQL

The orders table has been created using SQL, this table is where the order information is stored, such as the order ID, the customer ID and the book ID. All of this information is necessary to pick the book the customer wants, and send it to them.

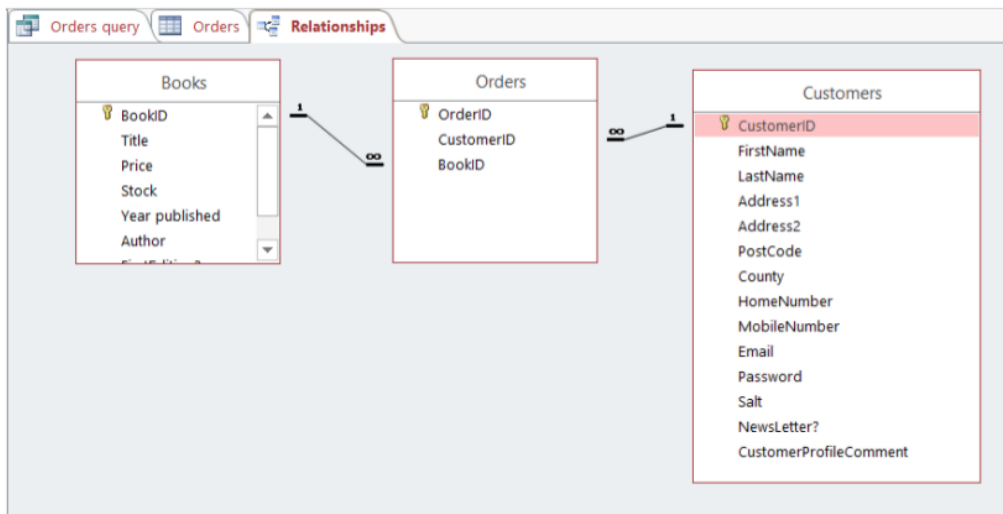
Create table creates the table. OrderID creates a field, int makes it an integer, not null means it cannot be left blank. This is the primary key and has been designated so with the primary key command. CustomerID and BookID create these respective fields.

To create the relationships between the two tables using SQL, it has been stated that the IDs are foreign keys, then it references which table and ID the relationship connects to. Clicking run then creates the table. There can be many orders to one book, and many orders to one customer.

```

Orders query
CREATE TABLE Orders
(
  OrderID int NOT NULL PRIMARY KEY,
  CustomerID int,
  BookID int,
  FOREIGN KEY (CustomerID) REFERENCES Customers (CustomerID),
  FOREIGN KEY (BookID) REFERENCES Books (BookID));
  
```

Orders query	Orders
OrderID	CustomerID
	BookID



Database Design (The Book Store)

This table can then be populated by creating another SQL query and using the insert into command, targeting the table name the user wishes to populate, then values, then the values. Finally the user must click run. This can be done multiple times by deleting the values and entering new ones to create multiple rows within the table.

Orders query Insert values into Orders table Orders

```
INSERT INTO Orders VALUES (1, 1, 1);
```

Orders query Insert values into Orders table Orders

```
INSERT INTO Orders VALUES (2, 3, 2);
```

Orders query	Insert values into Orders table	Orders
OrderID	CustomerID	BookID
1	1	1
2	3	2
*		

3.2.2 Coding queries within SQL

A query has been written to search for book titles using the book ID. It can be manually edited by changing the ID to a different number which selects the record with the corresponding number from the books table.

Search book ID

```
SELECT Title
FROM Books
WHERE BookID = 1;
```

Search book ID
Title
Jojo's Bizarre Adventure Part 1: Phantom Blood
*

3.3 Additional queries

A query has been created to create the orders table, this was coded within SQL and double clicking it creates the table. Values can then be inserted into the orders table by editing the “Insert values into Orders table” query and then running it.

A query is provided that searches for all book titles, and displays them all on the screen without having to look at the books table. Another query also pulls up every book ID. Another query has been created that looks up where every book is stored, such as which warehouse it is located in, and which isle and floor it is stored on so it can be easily collected and shipped.

A query exists that finds which customers have ordered which books. Finally, a query has been created to find which customers have subscribed to the newsletters so they may be mailed the newsletter with ease.

3.3.1 Producing a report from a query

The query used is one displaying all customers who have subscribed to the newsletter. A report is produced from this query by clicking the report command when the query is selected. This has been done with all other notable queries such as: Which customer has ordered which book? And where is each book stored? The layouts of each has been appropriately altered to allow all information to be fully displayed.

 Which customers have subscribed to the newsletter1

Which customers have subscribed to the newsletter			06 May 2020 03:12:57
CustomerID	FirstName	NewsLetter?	
4	Morthos	☒	
7	Tyrion	☒	
9	Samantha	☒	
11	Maybel	☒	
13	Sandy	☒	
15	The	☒	
1	Hol	☒	
7			

Page 1 of 1

3.4 Main Menu

The main menu was created by selecting a blank form and populating it with buttons designed to open forms and queries. This is so staff who aren't Database administrators can easily access information without needing access to tables. The main menu has also been coloured a light blue for visibility.

The main menu is displayed automatically when the Database is opened, this was done by going into options, then current database, then selecting the main menu as the display form.

Additionally, every form grants the user the ability to add or delete a record. This is so records can be added on the fly by staff without them needing administrator rights to alter the tables themselves.

4.0 Conclusion

A database solution for the book store has been provided that features a range of appropriate tables with relationships linking them together to allow for embedded information which streamlines the company system. Multiple queries have been created to narrow down specific information. A table and relationships has been created by code using SQL. A plethora of forms have been provided to navigate through information, and said forms allow for data to be added or deleted on the fly. A main menu has also been provided for easy access to information, and this opens automatically when the Database is opened. Repots have also been compiled from queries.

5.0 Appendices

Appendix A: SQL Code

Orders query:

```
CREATE TABLE Orders
```

```
(
```

```
OrderID int NOT NULL PRIMARY KEY,
```

```
CustomerID int,
```

```
BookID int,
```

```
FOREIGN KEY (CustomerID) REFERENCES Customers (CustomerID),
```

```
FOREIGN KEY (BookID) REFERENCES Books (BookID));
```

All book titles query:

```
SELECT "ID: " & BookID & ", Title: " & Title AS Test
```

```
FROM Books;
```

Search book ID query:

```
SELECT Title
```

```
FROM Books
```

```
WHERE BookID = 1;
```

Insert values into Orders table:

```
INSERT INTO Orders
```

```
VALUES (1, 1, 1);
```

FINAL GRADE

GENERAL COMMENTS

51 /100

Instructor

Generic Assessment Criteria Kallum Gordon 51%	Marks available	Marks awarded
1. Engagement with Literature Skills		
2. Knowledge and Understanding Skills		
<i>Suitable development methods employed and justified. Using the above scenario,Functionality</i> Although you have built a relationship diagram in Access there is no evidence of prior design considerations within Visio	25	10
3. Cognitive and Intellectual Skills		

<i>Provide evidence that the tables will link together in a relational database environment. For this section you should also provide evidence of the design techniques that you have used to develop the database system</i> <i>Understanding of design and SQL</i> You have linked your tables correctly and presented correct links adhering to referential integrity. I can see that you have placed evident of SQL in your report but it would have been better to show more of this within the Access environment.	20	11
4. Practical Application Skills		

Successful demonstration of software implementation. You are required to implement a database to meet the requirements, attribute field to enhance the design) Forms, Queries, Reports, Relationships, Macros A suitable but basic main menu and you have explored some if the available functionality within Access to build forms and reports. You have added some suitable test data. No macros used.	45	23
5. Transferable Skills for Life and Professional Practice		
All work suitable presented and referenced. Your work has been presented well and you report details your system build	10	7

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11
