|  |
| --- |
| **Database Assignment** |

1. **Introduction**

Uncle Johnny has a bookstore located at heart of a town in Singapore. Due to the high rental cost, Uncle Johnny has no choice to windup his business. In order to pursue his passion for operating bookstore, he has idea of setting an online bookstore. The online bookstore is named Fall 4 books. His idea is to encourage more people to read more books. The aim of the online bookstore is to reach out for more people with a low cost setup and so he wants me to advise him on this issue.

1. **Data Modelling**

To develop a data model for the system, I have to identify the starting entity types for the system. In my opinion, the following entity types are needed:

BOOK (**BookID**, BookTitle, BookType , BookPrice, BookAuthor, CatID, CatName)

CUSTOMER (**CustID**, CustName, CustAddr, CustTelNo, CustEmail)

ORDER (**OrderNo**, OrderDate, CustID, CustName, CustEmail, [BookID, BookTitle, OrderQty, BookPrice], SpecRequest, DeliDate, DeliAddr)

**Notes**: 1. Data items are highlight in bold and underlined indicate it is unique data item.

2. Data items which are in the [brackets] imply a repeating group within an entity type.

A data model was developed using entity relationship diagramming technique. The entity relationship diagram (ERD) is shown as follows:

Explanation of the entity-to-entity relationships:

1. A book may cater for one or more orders. An order must be catered by one and only one book.
2. An order must be placed by one and only one customer. A customer may place one or more books.
3. **Data Normalization**

To normalize the entity types, I transformed the starting entity types from un-normalized forms to third normal forms using a step-by-step transformation technique. The process is described as follows:

1. Transformation from Un-Normalized Form (0NF) to First Normal Form (1NF)

To transform 0NF to 1NF, I have to remove repeating groups to form new entity types. The resulting 1NF are as follows:

BOOK (**BookID**, BookTitle, BookType , BookPrice, BookAuthor, CatID, CatName)

CUSTOMER (**CustID**, CustName, CustAddr, CustTelNo, CustEmail)

ORDER (**OrderNo**, OrderDate, CustID, CustName, CustEmail, SpecRequest, DeliDate, DeliAddr)

ORDER DETAIL (**OrderNo**, **BookID**, OrderQty , Book price)

1. Transformation from 1NF to Second Normal Form (2NF)

To transform 1NF to 2NF, I have to remove partial key dependencies to form new entity types. The resulting 2NF are as follows:

BOOK (**BookID**, BookTitle, BookType , BookPrice, BookAuthor, CatID, CatName)

CUSTOMER (**CustID**, CustName, CustAddr, CustTelNo, CustEmail)

ORDER (**OrderNo**, OrderDate, CustID, CustName, CustEmail, SpecRequest, DeliDate, DeliAddr)

ORDER DETAIL (**OrderNo**, **BookID**, OrderQty , Book price)

BOOK-2 (**BookID**, OrderQty , Book price)

1. Transformation from 2NF to Third Normal Form (3NF)

To transform 2NF to 3NF, I have to remove non-key dependencies to form new entity types. The resulting 3NF are as follows:

BOOK (**BookID**, BookTitle, BookType , BookPrice, BookAuthor, CatID )

CATEGORY (**CatID**, CatName)

CUSTOMER (**CustID**, CustName, CustAddr, CustTelNo, CustEmail)

ORDER (**OrderNo**, OrderDate, CustID, SpecRequest, DeliDate, DeliAddr)

CUSTOMER-2 (**CustID**, CustName, CustEmail)

ORDER DETAIL (**OrderNo**, **BookID**, OrderQty , Book price)

BOOK-2 (**BookID**, OrderQty , Book price)

The starting entity types have been transformed from 0NF to 3NF. But some of the 3NF entity types have exactly the same unique identifiers. Therefore, they may be combined if their meanings are the same. After combining the 3NF entity types, I have the following remaining 3NF entity types:

BOOK (**BookID**, BookTitle, BookType , BookPrice, BookAuthor, CatID )

CATEGORY (**CatID**, CatName)

CUSTOMER (**CustID**, CustName, CustAddr, CustTelNo, CustEmail)

ORDER (**OrderNo**, OrderDate, CustID, SpecRequest, DeliDate, DeliAddr)

ORDER DETAIL (**OrderNo**, **BookID**, OrderQty , Book price)

1. **Data Model After Normalization**

After normalization, a new data model can be developed using the remaining 3NF entity types. This data model is a normalized ERD for the system, as shown below:

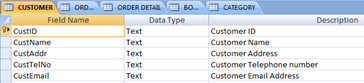
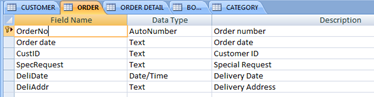
Explanation of the entity-to-entity relationships:

1. It shows that a category may classify one or more book and a book must be classified by one and the only one category.
2. It shows that one order must comprise one or more order books and an order must belong to one and the only one order.
3. It shows that a book may cater for one or more order books and an order must be catered by one and the only one book.
4. It shows that an order must be placed by one and only one customer and a customer may place one or more books.
5. **Design & Implementation of System Database**

In this process, the creation of the table are base on each entity type in the normalize data model. Primary keys in the tables must be identified according to each of the normalized entity types.

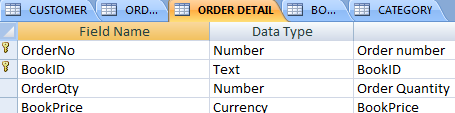
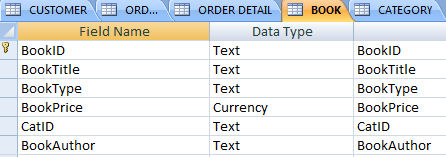
[1] Table : Customer [2] Table : Order

Primary Key: Cust\_ID Primary Key: OrderNo

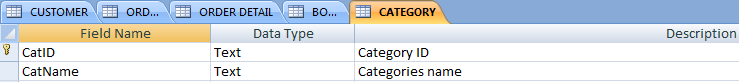
 

[3] Table : Order Detail [4] Table : Book

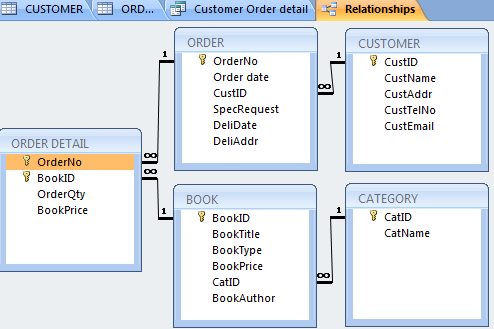
Primary Key: OrderNo Primary Key: BookID

** **

[5] Table: Category Primary Key: Cat\_ID

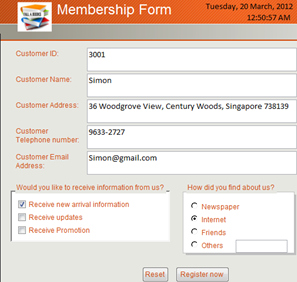
****

Relationships between the tables are defined and are link to the respective to the data as shown below:

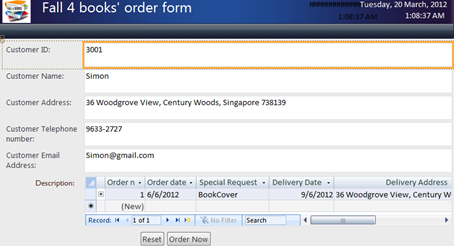
****

1. **Design & Implementation of System Forms**

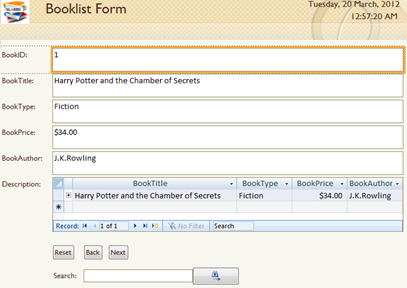
System Form is important the user as it is very convenience and useful to the user. There is a need to design a unique form and with the Fall 4 books logo on it indicate that it is belong to Fall 4 book’s online store .After analyses of the normalized data model, I have come up with design and implement forms for the system as shown below:

1. **Membership Form**

Members are free to sign up the membership form to become a member of the Fall 4 books.

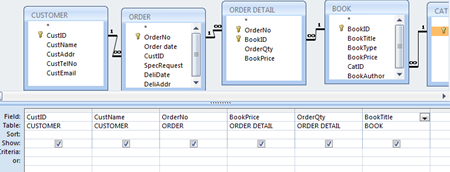
1. **Fall 4 books’ Order Form**

It show customer order detail with the order date, delivery date and delivery address.

1. **Booklist Form**

It shows the list of book with description of price, type, title and the author.

1. **Implementation of System Queries**
2. Uncle Johnny is able to create query to display important information which he needs by using the query function.



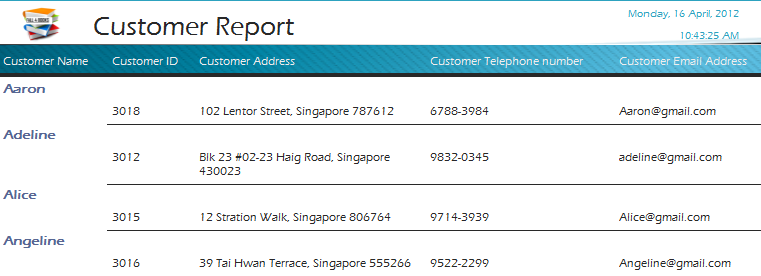
1. After Running the Queries, It will immediately display the data that is selected in the queries.



1. **Design & Implementation of System Reports**

The Design of this report have alternate highlight of colour for easily view of each individual’s data as shown below:

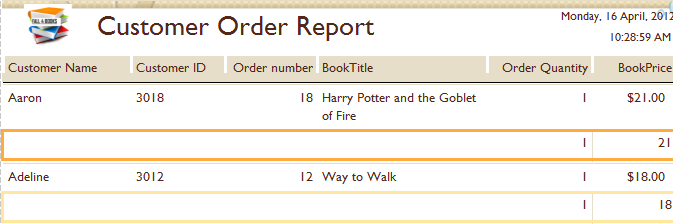
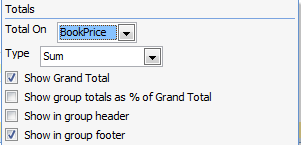
1. **Customer Report**
2. The customer report show the customer name sorted with alphabetical order



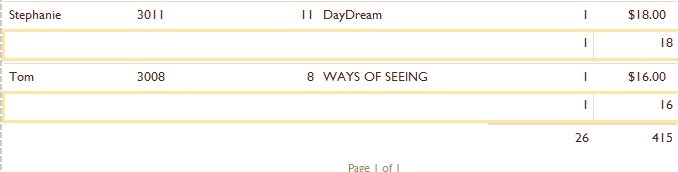
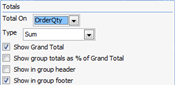
1. **Delivery Report**
2. The delivery report show the order date, delivery date and delivery address.



1. **Customer Order Report**
2. At the top of the report page, it shows subtotal of book price and quantity. There is screenshots beside it show how it is done.

1. At the end of the report page, it show the grand total of quantity and Book Price

1. At the bottom of the page, it shows that the customers have sorted.



1. **System Operations**

Finally, Uncle Johnny can now use the system. He will need to click on the database application with its file named “Fall4Books.DB”. The main page is shown as below:

The main page of the system contains information of the Form, Queries and Report.

Uncle Johnny can use the system by clicking on the options provided under each of the categories.

1. **Conclusions**

This Database assignment gives me an opportunity to explore the database feature. During the assignment, I have made quite a lot of mistakes. For example, the data type for the book price should be in currency and not identified as a number or a text which is crucial when doing subtotal and grand total in the customer order report. If the data type of the book price is a text, it is unable to sum all together and it is a waste of time to change the data type. After the mistake that I have made, I have learnt to become wiser and able to work efficiently using Microsoft Access.