

Computer Science 1600
Assignment 4 (Distance Section)
Spring 2019
Due on Tuesday, July 30, 2019,
before the cut off time of 11:59pm (Newfoundland Time)

Associated Lab Work

Lab 6, 7 and 8

File to be used with this assignment: [COMP1600_A4_S19_DESIGN.docx](#)

PLEASE NOTE: You should read the ENTIRE assignment before beginning to work on it. There are details throughout the assignment regarding the requirements for this assignment.

Part I – Designing the database

Read Again Books is a second hand book store which has a large collection of books of different genres. The manager, Ms. Brooks wants to keep track of the books and would like a database to maintain the information about the books, the authors and the publishers of the books. You are asked to design the database for Ms. Brooks. Some of the information that she wants maintained in the database is as follows:

Title: the book's title

ISBN: the 13-digit International Standard Book Number that uniquely identifies a book

PName: the name of the publishing company that published the book

LPrice: the list price of the book, i.e., the approximate price of the book when sold new

SPrice: the sticker price of the book, i.e., the price at which this store sells the second hand book

PYear: the year that the book was first published

Pages: the number of pages in the book

Genre: the type of book it is, i.e., Biography, Travel, Fiction, etc.

Notes: the manager's comments about the book which, in general, the customer's find very helpful

AName: the first and last name of the author (**please note that more than one author can collaborate to write a book**)

AGenIdentity: since some readers may be looking for books by authors based on their gender identity, this field should hold the current gender identity of the author, if specified (sample values may include, but are not limited to, female, male, transgender, non-binary, unspecified, etc.)

ACountry: since some readers may be looking for books by authors from a particular country, the country of origin for the author is also included

Awards: indicates if the book has won any awards or not (yes or no)

SRating: the star rating of the book from 1 to 5, with 5 being the best

On Shelf Date: the date when the book was placed on the shelf to be sold

Off Shelf Date: the date when the book was sold and hence taken off the shelf (if not sold, date is 12/31/1900)

PCity: the city where the publisher has its head office

PCountry: the country where the publisher has its head office

PPhone: the publisher's contact number

PEmail: the publisher's e-mail address

In addition to the data listed above, you may need to add other fields/tables depending on the types of queries that may be asked; hence please keep the following in mind when designing your database. Ms. Brooks wishes to see the following information from the designed database (**PLEASE NOTE THAT YOU DO NOT NEED TO IMPLEMENT THESE QUERIES FOR THIS ASSIGNMENT**):

1. List the title of each book written by Jilly Cooper.
2. List the book titles and author's last name, followed by her/his first name, for the books in the Spiritual genre.
3. Answer the question: Does the store have a book called "Riders" that was published in 1978?
4. List the name and the country of the publisher who published the book "Bunch of Grapes".
5. List the name of the book, and the year that the book was published, for all the books authored by female authors.
6. List each book's title, sticker price, number of pages and rating of all the books that are currently on the shelf. Your list should be sorted in ascending order of the book title. **Hint:** Unless the book is sold and is actually off the shelf, the default value for the *Off Shelf Date* field would be 12/31/1900. If it is any other date beyond 12/31/1900, then it means that the book was sold on that date and is actually off the shelf.
7. List the title of the book, the publisher's name and the publication year for each book written by Agatha Christie. Your list should be sorted by the title of the book in ascending order.
8. A listing of all the books which are published either by Penguin Books or by Bantam Books. Your list should include the title and the rating of the book, as well as the first name, followed by the last name, of the author. Your list should be sorted by publisher's name and within the publisher's name by the book title, both in ascending order.
9. A list of all the books that have been on the shelf (and are not currently sold) for more than 30 days. Your list should include the title, the rating, the author's first name and last name, and the number of days that the book has been on the shelf. Your list should be sorted in the descending order of the number of days that the book has been on the shelf.
10. A listing of all the books that were published between 1930 and 1950. Your list should include each book's title, genre, rating, whether the book won an award or not, the year the book was published and the publisher's name.
11. List each publisher's name and the number of books each publisher has published. In addition, list the total, the average, the highest and the lowest list price of the books from each publisher. The list should be sorted by publisher's name in ascending order. All columns should be appropriately labeled.

Provide a **design** of the database with normalized tables by completing questions a-e given below. [Type your design into the given MS-Word document, called COMP1600_A4_S19_DESIGN.docx.](#)

While designing the database keep in mind the issues of redundancy and duplication of information, it will be necessary to incorporate more than one table in your design. Any table or field that you add should have meaningful names.

Not all the information needed in the database has been specified above. Design decisions of inclusion or exclusion of information will be made by you, the designer, after considering the information requirements of *Read Again Books*.

- a). List the table names and the description of each table/entity that will exist in the completed database by filling in the table given below. Be sure to indicate if a table has been included as a relationship table to implement a many-to-many relationship. For example, for a clinic with many doctors (some specializing in a particular health field) and many patients, where each doctor can see any of the patients and each patient can get an appointment with any of the doctors, the database could be as follows:

Table Name	Description
Doctors	Represents information about doctors
Patients	Represents information about patients
Visits	Relates the two tables Doctors and Patients and maintains information regarding patients visits and which doctor they saw during that visit
Specializations	Maintains information about the specialization of health fields offered at the clinic

Complete the following table (in the design document) for the given problem (you may not need all rows in the table):

Table Name	Description

- b). Give the tables/entities and their relationships (in tabular form) indicating the cardinality (also known as multiplicity) of the relationship from **Entity 1** to **Entity 2** by completing the table given below. For example, consider a database for a walk-in clinic where a patient can be seen by any doctor, and a doctor can see any of the patients.

Entity 1	Entity 2	Cardinality of Relationship From Entity 1 to Entity 2 (1:M, M:1 or 1:1)
Patients	Visits	1:M
Doctors	Visits	1:M

Complete the following table (in the design document) for the given problem (you may not need all rows in the table):

Entity 1	Entity 2	Cardinality of Relationship From Entity 1 to Entity 2 (1:M, M:1 or 1:1)

- c). List each table in the database, including the name of each table with its attributes listed within parenthesis, **with the primary key being the first field listed and being bold/underlined**. The following notation should be used:

Doctors(**SIN**, Firstname, Lastname, Address, Phone Number, SpecialtyID)

- d). Using the **From/To** table given below, indicate how the primary and foreign key fields of the tables in the database are used to connect/join the related tables. For example,

Specialization(**SCode**, Description of Specialty)

Doctors(**SIN**, Firstname, Lastname, Address, Phone Number, SpecialtyID)

From Table	From Field	To Table	To Field
Specialization	SCode	Doctors	SpecialtyID

Complete the following table (in the design document) for the given problem (you may not need all rows in the table):

From Table	From Field	To Table	To Field

- e). Provide the structure of the **Books** table only (including the field name, data type, size and description), in the manner shown here:

Table Name: Patients

Name	Type	Size	Description
PatientID	Text	10	Primary key that uniquely identifies a patient
Phone	Text	10	Contact phone number

Complete the following table (in the design document) for the given problem (you may not need all rows in the table):

Name	Type	Size	Description

Part II – Implementing, modifying and using the designed database

- a). Using **MS-Access**, create a file called **COMP1600_A4_S19_IMPLEMENT.accdb**. In this file, create the structure of all the tables for your database and add data to the tables (at least 5 records in each table where possible; **the data is to be made up by you**). The data you enter should be such that you could answer the potential queries listed above; **however, these queries do NOT have to be implemented for this assignment.**
- b). Secure your database from incorrect data entry by adding validation rules/texts for each of the following to your **COMP1600_A4_S19_IMPLEMENT.accdb** file:
- The owner of the bookstore has decided not to carry books that are longer than 1500 pages. Include a validation rule and validation text to make sure that the books are no more than 1500 pages in length.
 - In the **Books** table, no **Off Shelf Date** should be prior to the **On Shelf Date**, unless the book is still on the shelf (i.e., if the **Off Shelf Date** is 12/31/1900). To do this, it seems like we might be able to just enter a validation rule for the **Off Shelf Date** that compares it to the **On Shelf Date**. Unfortunately, this won't work. **MS-Access** doesn't allow us, at the column level, to compare two columns. However, we can add a validation rule at the table level. When the **Books** table is open in **Design View**, click on **Property Sheet** on the ribbon. This will open the property sheet for the table. Here we can enter the validation rule:
[Off Shelf Date]>=[On Shelf Date] Or [Off Shelf Date]=#12/31/1900#
and validation text:
Off Shelf Date must come after On Shelf Date or be 12/31/1900.
 - For the star rating field (**SRating**), make sure that the star rating is between 1 and 5, inclusive.
 - Test each of your validation rules to make sure they work by entering incorrect data. **Capture the Validation Text message displayed (using the Alt-PrintScreen keys) and paste it in your MS-Word document called COMP1600_A4_S19_DESIGN.docx.**
- c). Using the **Relationships** tool, establish relationships between the tables (**enforcing referential integrity** each time), as shown in your database design. Once the relationships are established, try to delete an **Authors** record, **for an author that has written at least one book**. **Capture the message displayed (using the Alt-PrintScreen keys) and paste it in your MS-Word document called COMP1600_A4_S19_DESIGN.docx.**
- d). Create a **form** to be used to add new records to the **Books** table (called **Add a Book**). Add a few extra order records to your **Books** table, using this form.
- e). Create an inventory report for the book store to include a list of all the books from different genres that are currently **on the shelf**. Your report should include each type of genre (i.e., the genre description), and the title and sticker price of each book. Your report should group the listing by genre and show the total sticker price for all books under that genre. **Note:** You may or may not have to base this report on a query which you would create before generating the report. **You should include enough data in your database to sufficiently test this report.**

Submission

Submit your **COMP1600_A4_S19_DESIGN.docx** file and your **COMP1600_A4_S19_IMPLEMENT.accdb** file containing your solution using Brightspace's **Dropbox** tool. Note that the Brightspace **Dropbox** has been set up to allow you to submit updates to your assignment multiple times until the due date/time.