A·P·U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

GROUP ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

AICT005-4-1-DAS
DATABASE SYSTEMS

UCDF2005(1) ICT(DI)

ASSIGNMENT TITLE: APU'S E-BOOKSTORE (PART 1)

GROUP MEMBERS: LEE DONGXUAN (TP059509)
KOK HON KIT (TP059378)
YAM YIN YEE (TP060340)

LECTURER : SEETHA LETCHUMY

HAND OUT DATE : 23 - MARCH- 2021

HAND IN DATE : 18 -MAY- 2021

Table of Contents

1.0 Introduction

This report is to show a database solution using SQL Server based on a case study about APU’s E-Bookstore. This report provides disadvantages of file-based system. This report also provides advantages of database and database management system (DBMS), and functions of DBMS while being implemented in the case study. This report also generates a list of business rules in the case study. This report also performs normalisation up to third-normalised form (3NF) on an unnormalized form (UNF) according to the case study. The report also provides the Entity Relationship Diagram of the database designed by the group which will be implemented in SQL Server later.

2.0 Database and Database Management System (DBMS)

2.1 Disadvantages of file-based system

File-based system is a collection of application programs to provide users the service of accessing information. In file-based system, each program used to manage its own data. Thus, limitations and restrictions may be occurred, which will lead to number of drawbacks of file-based system. From the case study, APU's E-Bookstore, it shows current APU library and small bookshop saved the books and reading materials in the form of file-based system. Number of disadvantages regarding this system shown, such as security problems and difficult to accessing data. Those disadvantages could result wrongly, undated information transferred to users who access to certain places. Below (Figure 2-1) are the major drawbacks of file-based system:

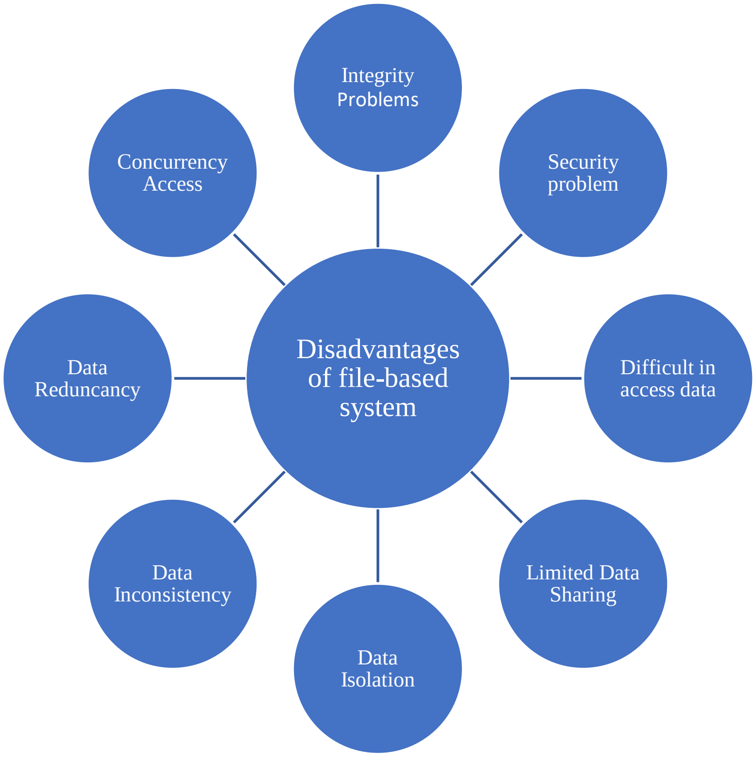


Figure 2-1: Disadvantages of File-based system

Data Redundancy occur when multiple copies of the same data appeared, like example in (Figure 2-2). In file-based system, this problem came in often which the management of the file can be easily read. On the other hand, it may lead to a lot of storage will be wasted from the duplicated of data (Tutorialink, n.d.). Beyond the memory wastage, data redundancy may also lead to **data inconsistency**, which defined as data stored in different files may not be consistent state. According to file-based system, it is difficult to update the information in every single duplicated file. Thus, it will result as every file could have different information. Based on the case study, APU Library provide a collection of books in the form of hardcopy and e-books and even a small bookshop. In this case, many duplicated information placed in different places, such library and bookshop. Whenever students and staffs will look for the files, many different information will be found, and it will lead to wrongly and outdated information transferred. In addition, they need to choose out which files can be trusted.

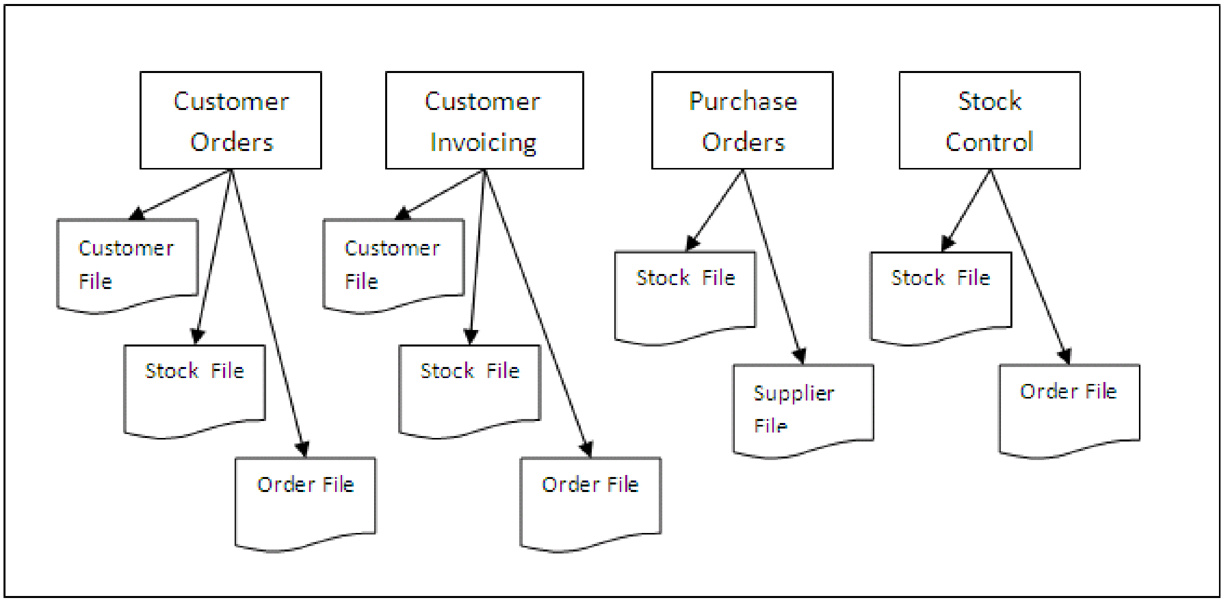


Figure 2-2: Examples of data redundancy

Data isolation defined as related data is separate stored in different files. It is cause by every program have different usage and formats of the data and it only manages its own data. Therefore, the users of one program may be accidentally not written some potentially useful data held by other programs. It will result as some other related program will be difficult to update and retrieve the appropriate data.

Limited Data Sharing is one of the disadvantages of file-based processing system. It is caused by each application program used its own private data files and it decentralized in various files, different formats, and departments. Furthermore, this processing system did not have any facility to share data file among multiple users. Thus, it will lead to **accessing data** is quite **difficult** and it is not convenient and efficient. For example, from the case study, students, and staffs in

APU only have the limited option of books can be choose in small bookshop and APU Library, which limited data sharing occurs in this case. Thus, students and staffs will be found difficult on searching the data they needed.

In files-based system, data should be stored in the manner of consistent and accuracy. On the other hand, to achieve this manner, **integrity problems** occur, which data records must satisfy correct and in the form of certain constraints (Tutorialink, n.d.). For example, in APU case study, the constraints may be TP Number of the students and staffs and which it could not be empty to specified in the application programs. Thus, it will lead to difficulty on making changes to the application program when enforce new constraints.

Concurrency Access to data is the ability of allowing multiple users access to the same data at the same time. In file-based system, concurrent access does not been allow. In addition, a file-based system should be control and prevent concurrency to be used by the application program (Watt, n.d.). Whenever an application had opened a file, others will not allow to access this file, because it will be locked to others. Based on the examples from case study, when student A and student B wanted to borrow the same book from APU Library, they access the application and check for availability at the same time. Sooner, the result will show it was available, but when they get the book, it will be marked as zero-copy in the file or no books available (TutorialCup, 2021). So, in this case, problem of concurrent access occurs.

Security Problem is the most baleful problem seem in file processing systems. In this system, people who can access to the files processing system, they can easily modify, and change data stored in those certain files. Thus, most of the management of the file used some restrictions in order to protect the files. Based on the case study, APU had also try to solve the security problems by setting a lot of restrictions. But, on the other hand, this had caused the students and staffs have limited options of books can be read in the library. It had also directly affected them on difficult to accessing the data from the Library.

2.2 Advantages of Database and DBMS

There are many advantages that can be brought by using database and Database Management System (DBMS). Compared to organize data using File System, Database and DBMS is more efficient in storing, retrieving, and updating data due to the functionalities provided in DBMS. The advantages of Database and DBMS are as below:

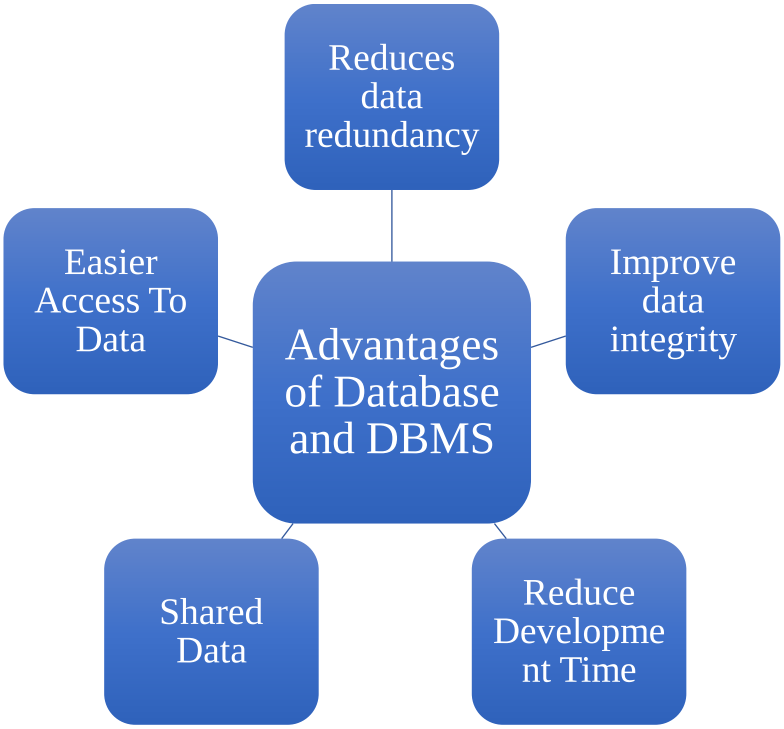


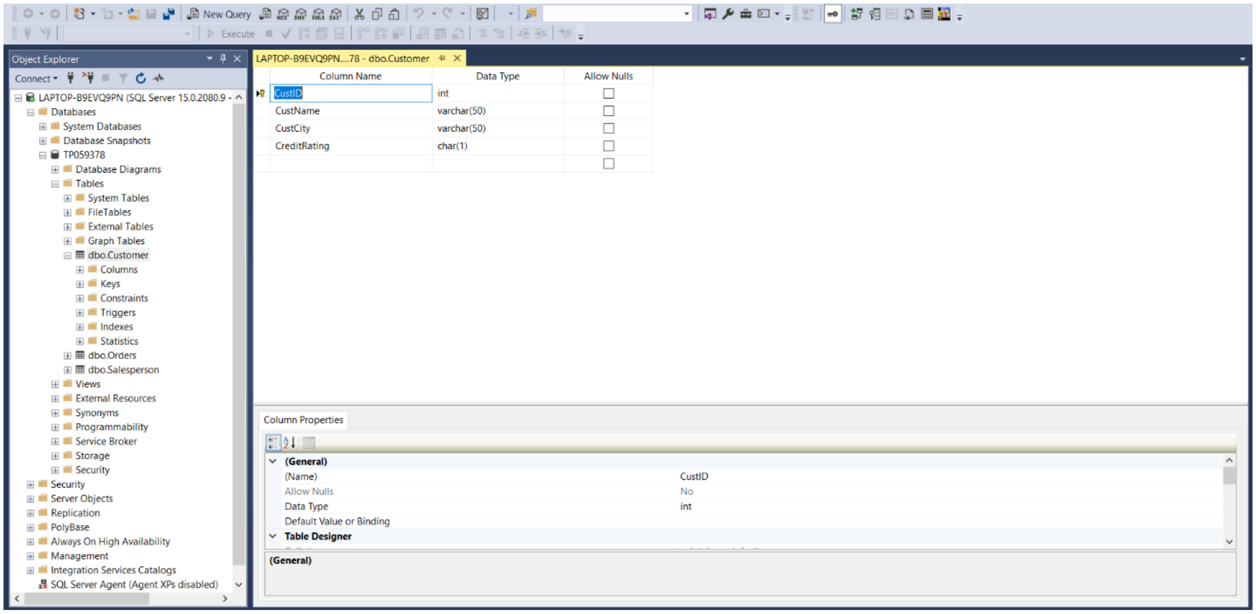
Figure 2-3: Advantages of Database and DBMS

By using file system to store the data, the data has higher probability to repeat which will cause data redundancy to happen. This is due to the reason that different departments are creating their own file system to store data. In this case, some of the data might repeat and will reduce the reliability of the data. As a result, data redundancy happens. Therefore, we can **avoid data redundancy** which results in data inconsistency by using Database and DBMS. This is because the DBMS introduces unique keys like primary key and foreign key. These keys are the unique column in the table of database which contain a unique value to prevent the same data from repeating. Besides that, a DBMS reduces data redundancy by limiting the isolated files where the data is repeated. This may not eliminate data redundancy, but it can greatly reduce the occurrence of data redundancy (Keneth C. & Jane P., 2006). From the case study, data sent by the publishers such as book title, authors and publisher name will be labelled a unique ID which is also known as primary key so that the data redundancy can be controlled.

The second advantage of the Database and DBMS is that it can improve **data integrity**. This can be achieved by setting the metadata of the fields in the database. For example, in a numerical column, the string data type cannot be inserted into the column. As a result, the data received is unaltered from when it was created and inputted to the database. Therefore, the accuracy and fidelity of the data is ensured in the database (Teeling, 2012). Aside from that, less errors are also made by human such as inputting the false data type which will affect the data integrity of the database. From the case study, if the APU's E-Bookstore uses database then the data integrity can be ensured due to the functionalities provided in the database. The functionality such as data dictionary management defines the metadata of the data and their relationships to improve the data integrity. For instance, the author and publisher name column can only receive string data type instead of integer data type to prevent input error from end users.

Figure 2-4: SQLite in Python (SQLite Tutorial, 2021)

The third advantage of using Database and DBMS is the Database and DBMS also help to **reduce the development time** of an application. This is because file management by using programming language to record data is a complicated process and it takes a long period of hours to successfully develop the application. Besides that, errors are also likely to occur when using file system to record data. However, most programming languages has framework that can support connections to the Database and DBMS which can greatly reduce the development time. For instance, the “sqllite3” library in Python can help the developer to easily connect their applications to the Database (Figure 2-4).



Column Name	Data Type	Allow Nulls
CustID	int	□
CustName	varchar(50)	□
CustCity	varchar(50)	□
CreditRating	char(1)	□
		□

(General)	
(Name)	CustID
Allow Nulls	No
Data Type	int
Default Value or Binding	

Figure 2-5: Graphical User Interface (GUI) of Microsoft SQL Server Management Studio 2018

On top of that, Database and DBMS are **easy to learn and use**. This is due to the reason that most DBMS uses language like Structured Query Language (SQL) to act as a bridge between the Database and the users. This language is very similar to English, so it is also friendly to those who are not familiar with programming knowledge. On the other hand, the DBMS has provided different pre-defined functions so that the user will not need to write the code from scratch. For instance, the Graphical User Interface (GUI) that the Microsoft SQL Server Management Studio 2018 provided (Figure 2-5) greatly reduce the complexity because the users can just create, retrieve, update and delete data by clicking buttons that are provided. From the case study, the staff at APU’s E-Bookstore can quickly learn to use the DBMS provided due to its simplicity.

After that, the Database and DBMS also allows the **sharing of data** among users. For instance, the organization integrates its data and information within a single database system. This allows the data to be shared among employees and others who have the authority to access the system from all different departments. This feature of database system allows the data to be shared among users more easily and conveniently. Besides that, this feature also gives the users the ability to generate more information from the data due to the integration of all the data within an organization (WATT, 2014). From the case study, the staffs within the APU’s E-Bookstore can share the data such as the sales, number of books received from the publisher and the number of members within the database so that decision making can be made more efficiently and effectively.

Aside from that, the advantage that the Database and DBMS brought is that the data is **easier to access** than the file system. This is because the DBMS provides different functions such as SELECT, COUNT, AVG and much more that the users can use to access different kind of data. This greatly reduces the time needed to find the targeted data in a large database. For instance, the staff from APU’s E-Bookstore can quickly search books that are in specified genre in a large database with a few SQL commands.

2.3 Functions of DBMS

DBMS is a multi-functional system in ensuring data integrity and consistency. The functions of DBMS can be summarised in the mind map (Figure 2-6) below.

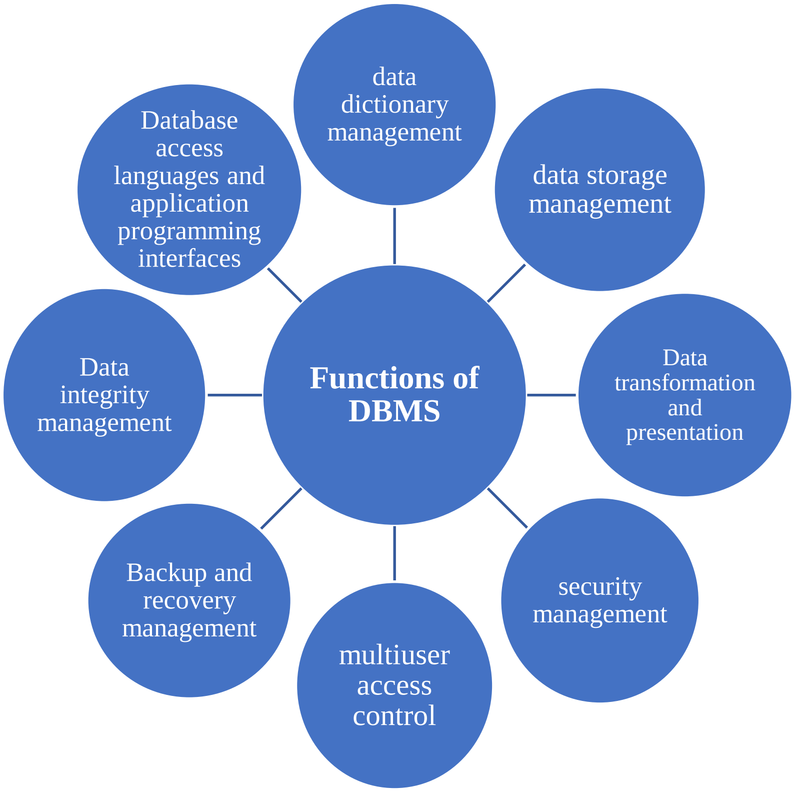
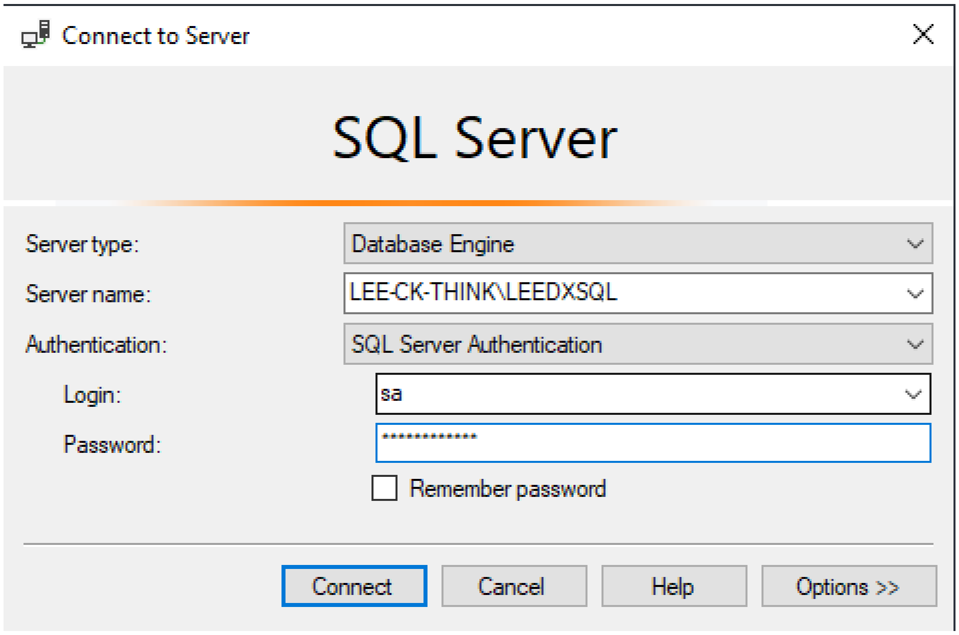

Functions of DBMS
data dictionary management
data storage management
Data transformation and presentation
security management
multiuser access control
Backup and recovery management
Data integrity management
Database access languages and application programming interfaces

Figure 2-6: Functions of DBMS

Firstly, DBMS is useful in **data dictionary management**. A data dictionary is a collection of names, definitions and characteristics about data elements that are being accepted and processed in a particular database (The Regents of the University of California, n.d.). In DBMS, a database designer defines which type of data the database can accept, the metadata of the data elements, and the relationship between them. For example, the designer of the e-bookstore's database defines the second column in the list of books sold in the e-bookstore as the name of a book, only short text is accepted by the DBMS, and there is a link between book list and sales record. Furthermore, the database designer can also define the metadata. In this scenario, he can define that 256 characters is the maximum characters accepted by the DBMS to be input into the column.

Besides, **data storage management** is one of the DBMS functions. This function is used to keep data and related data entry forms, report definitions. For instance, this function in the e-bookstore database is able to keep the book information, publishers' information, members' information and members' feedback. It can also keep reports regarding the rating of a particular book. This function also allows the DBMS to store data with different format such as tables and pictures. In the case study, the DBMS can store the cover page of the books under the datatype of image.

Data transformation and presentation is a function that translates logical requests into commands to physically locate and restore the requested data. In this scenario, the user has requested to select to view the top thousand records of books information from the database. This function translates this request into SQL commands in order to return the book information in form of table.



Connect to Server

SQL Server

Server type: Database Engine

Server name: LEE-CK-THINK\LEEDXSQL

Authentication: SQL Server Authentication

Login: sa

Password: *************

□ Remember password

Figure 2-7: Login with username and password

Moreover, **security management** is important to enforce user security and data privacy within database. It grants different access privileges to different types of users, that means it allows different users to access different types of data. In SQL Server Management Studio (SSMS), this function provides mixed mode authentication, it requires the user to undergo Windows authentication or SQL Server authentication in order to connect the server he wants to access. If he undergoes SQL Server authentication, he needs to provide the username which is “sa” and the password as shown in Figure 2-7. In the case study, this function is utilised to grant access to publishers to view and edit the book details. At the same time, it also gives access to members so that they can view the book details only.

Apart from that, **multiuser access control** is also one of the DBMS functions. It uses sophisticated algorithms to guarantee multiple users can access the database simultaneously without compromising data integrity. For example, assume that two members of the APU e-Bookstore are ordering the same book at the same time. However, there is only one book on stock and the publisher has not renew the stock that required by the two members in a short time. Without the concurrency control, both members purchase the book successfully, but only one member is able to get the book in a short time. With concurrency control, only one member who has completed the transaction process first is able to purchase and get the book in a short time.

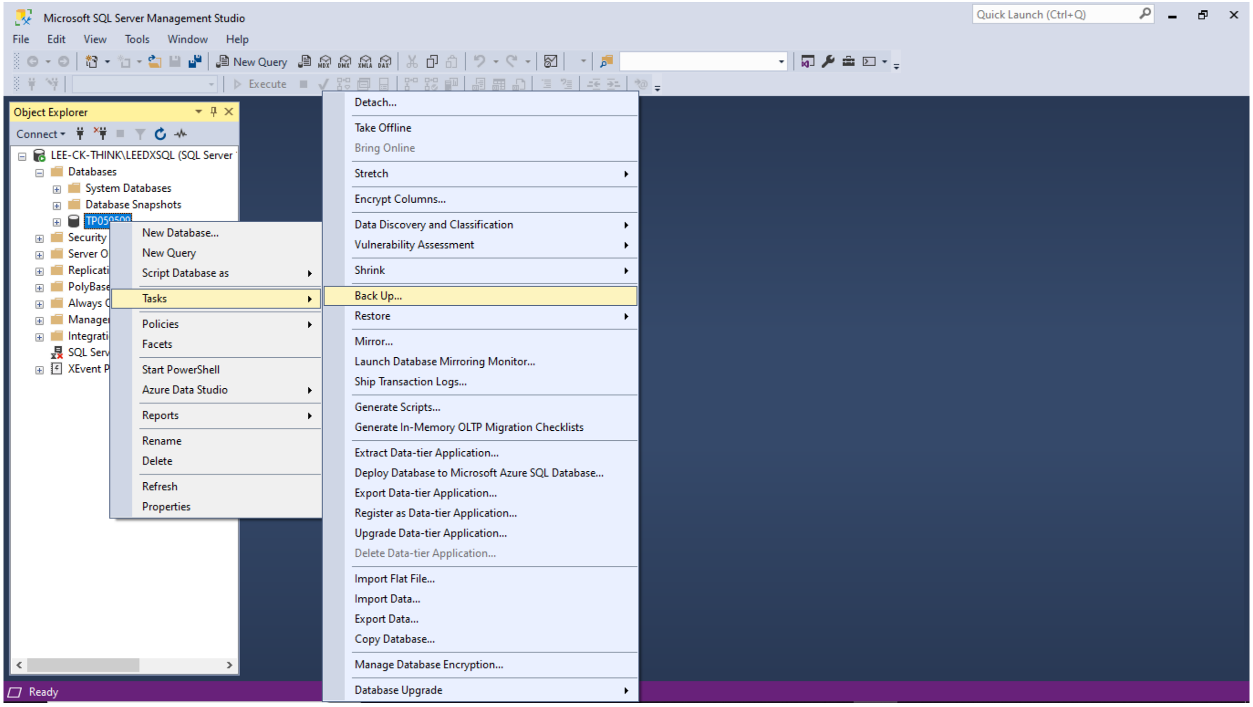
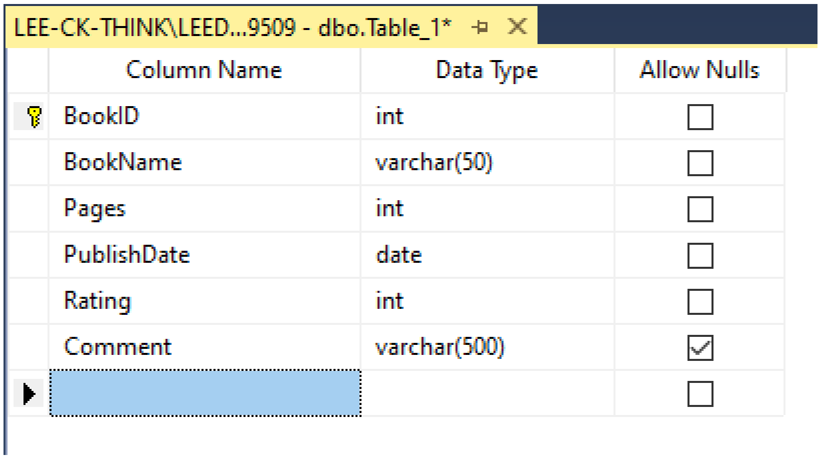


Figure 2-8: Backing up a database

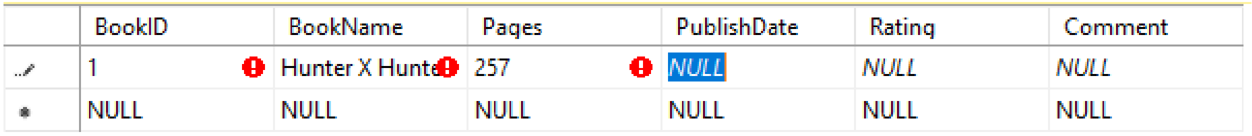
Backup and recovery management is a function that provides backup and data recovery procedures. As shown in Figure 2-8, it allows the user to duplicate a database in a different location. It is a very important procedure to maintain data integrity because it can prevent unexpected risks such as power outage, computer crash, system crash, and natural disasters such as hurricane, earthquake, and flood. In the case study, the database administrator should back up the e-bookstore's database regularly to avoid any data loss. If any unexpected risk occurs, he can recover the database in a short time.



LEE-CK-THINK\LEED...9509 - dbo.Table_1*

Column Name	Data Type	Allow Nulls
BookID	int	☐
BookName	varchar(50)	☐
Pages	int	☐
PublishDate	date	☐
Rating	int	☐
Comment	varchar(500)	☑
		☐

Figure 2-9: Unchecking the checkbox of the "Allow Nulls" statement



	BookID	BookName	Pages	PublishDate	Rating	Comment
	1 ❗	Hunter X Hunte❗	257 ❗	*NULL*	*NULL*	*NULL*
	NULL	NULL	NULL	NULL	NULL	NULL

Figure 2-10: Showing exclamation marks

Other than that, **data integrity management** helps in promoting and enforcing integrity rules. The completeness of data is very important to be maintained to avoid compromising the data integrity. Without data integrity, the data is said to be useless for analysis. For example, before allowing the end users to input data, the database designer needs to design the table. In order to maintain data integrity, he unchecks the "Allow Nulls" statement for the columns "BookID", "BookName", "Pages", "PublishDate", and "Rating" as shown in Figure 2-9. Generally, he does not allow the end users to left the column empty because this information is compulsory to be inputted. If the end user lefts the "PublishDate" and "Rating" column empty, the rest of the columns of a particular record which are not allow nulls will show the exclamation marks to warn the user to input the data (Figure 2-10). Else, it will cause an error and an error message box will pop out to stop the user from continuing inputting the data.

Database access languages and application programming interfaces provide data access through a query language. In the DBMS, an end user can utilise this function to use SQL statements to perform some actions. For instance, a customer wants to find a book with a rating of 10. Therefore, he opens a new query to filter out the books with rating of less than 10. After displaying the data, he can view a list of books with a rating of 10.

3.0 Business Rules & Normalisation

3.1 Business Rules in APU's E-Bookstore

The business rules are listed as below:

- Each **member** has an ID, name, gender, phone number and address.
- Each **book** has an ID, title, author, genre, price per unit, and quantity in stock.
- Each **feedback** contains ID, rating, and remarks.
- Each **shopping cart** has an ID, quantity bought, and total price by derivation.
- Each **order** has an ID, payment date and delivery date.
- Each **publisher** has an ID (identifier/key), name, address, phone number.
- Each **purchase** has its ID, quantity delivered, and date delivered.

Relationship:

- Each **member** can give many **orders**, and each **order** is given by a **member**.
- Each **member** has a **shopping cart**, and each **shopping cart** belongs to a **member**.
- Each **member** may give none or more **feedbacks** of their purchased books, and each **feedback** is given by a **member**.
- Each **shopping cart** contains one or many **books** and each **book** can be kept in one or many **shopping carts**.
- Each **shopping cart** can be converted to many **orders**, and each **order** is formed by a **shopping cart**.
- Each **order** contains many **books**, and each **book** belongs to many **orders**.
- Each **book** may have none or more **feedbacks** which given by members, and each **feedback** belongs to one **book**.
- Each **book** must have one **publisher**, and each **publisher** publishes one or many **books**.
- Each **book** belongs to many **purchases**, the **purchase** can contain many **books**.
- Each **publisher** can supply many **purchases** to the bookstore, and each **purchase** is supplied by a **publisher**.

3.2 Normalization

Unnormalised Form (UNF)

Table: ORDER

OrderID	paymentDate	deliveryDate	memberID	customerName	phoneNumber	bookID	title	PricePerUnit	QtyBought
1001	14/4/2021	20/4/2021	M001	Lee Dongxuan	011-5102321	101	The Little Prince	25.00	2
						102	The Hobbit	50.00	1
1002	25/5/2021	27/5/2021	M002	Kok Hon Kit	011-4326579	101	The Little Prince	25.00	3
						104	A Time to Kill	35.00	2
1003	8/11/2020	13/11/2020	M003	Natasha binti Ali	013-3485538	102	The Hobbit	50.00	1
						105	Star Trek	26.00	3
1004	7/5/2021	11/5/2021	M004	Haarith bin Aaqil	014-2934056	103	The Jungle Book	22.00	2

1005	9/6/2021	13/6/2021	M005	Siva a/l Harrish	011-45763948	101	The Little Prince	25	4

ORDER (OrderID, paymentDate, deliveryDate, memberID, customerName, phoneNumber, **(bookID, title, pricePerUnit, QtyBought)**)

In UNF, the table has repeating data such as bookID, title, pricePerUnit and QtyBought has repeated. To proceed to 1NF, each row must have only 1 value and each record must be unique. Therefore, a composite key must be identified to ensure that every record is unique.

1NF

ORDER (OrderID, bookID, paymentDate, deliveryDate, memberID, customerName, phoneNumber, title, pricePerUnit, QtyBought)

Functional Dependencies

OrderID → paymentDate, deliveryDate, memberID, customerName, phoneNumber (partial dep)

bookID → title, pricePerUnit (partial dep)

memberID → customerName, phoneNumber (transitive dep)

After flattening the table, the composite key identified consists of OrderID and bookID. Besides, a few functional dependencies are identified. Partial dependencies are found because the OrderID and bookID can represent some of the other non-primary key columns. Transitive dependency is found because a non-primary key column such as memberID can represent other non-primary key columns. To proceed to 2NF, the partial dependencies must be removed so that every table has only 1 primary-key column.

2NF

ORDER_DETAILS (OrderID, paymentDate, deliveryDate, memberID, customerName, phoneNumber)

BOOK (bookID, title, pricePerUnit)

ORDER_2 (OrderID, bookID, QtyBought)

After removing partial dependencies, the ORDER table is separated into 3 different tables with their respective primary key column. The records of ORDER_DETAIL are represented by OrderID, records of BOOK are represented by bookID while records of ORDER_2 are represented by OrderID and bookID. To proceed to 3NF, the transitive dependencies must be removed.

3NF

MEMBER (memberID, customerName, phoneNumber)

ORDER_DETAILS_2 (OrderID, paymentDate, deliveryDate, memberID)

BOOK (bookID, title, pricePerUnit)

ORDER_3 (OrderID, bookID, QtyBought)

After removing transitive dependencies, a new table MEMBER is created, and its records are represented by memberID. Meanwhile, the customerName and phoneNumber are removed from the ORDER_DETAIL table to avoid data redundancy.

4.0 Entity Relationship Diagram

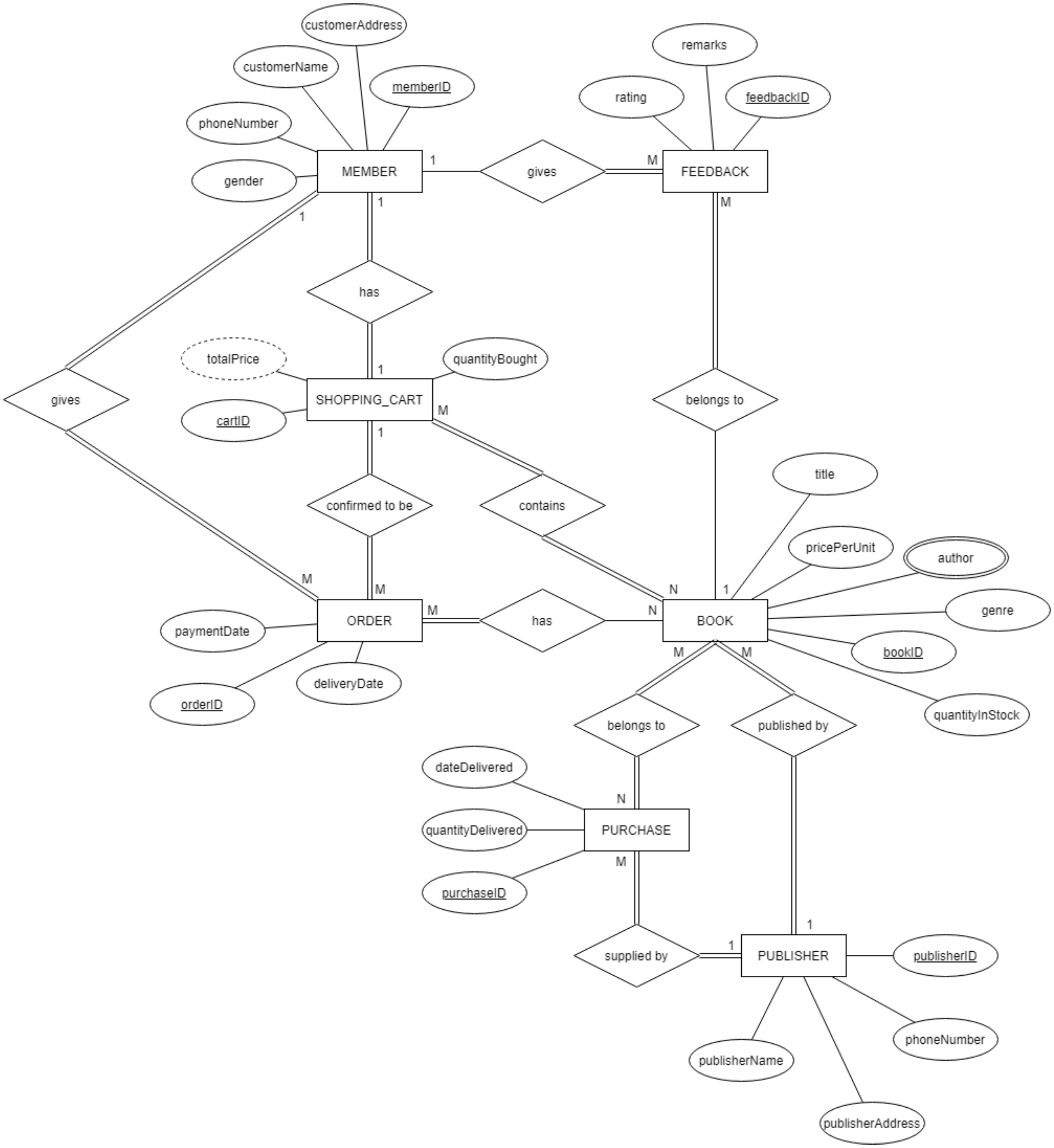


Figure 4-1: Entity Relationship Diagram

5.0 Conclusion

In a nutshell, this report discussed three main keys which included **database and database management system**, **business rules and normalization** and **entity relationship diagram** based on the APU’s E-bookstore case study given. To finalized, DBMS is more suitable to use in APU’s E-bookstore compared to file-based system, as it is more efficient and wider area on retrieving and storing data. In addition, business rules and normalization had been generated in order to develop soe form of constraints on the database. Lastly, the entity relationship diagram had also been drawn with the respective of their attribute, entities and relationship identified from the business rules.

6.0 References

Keneth C., L. & Jane P., L., 2006. *Management Information System Chapter 7.* [Online]
Available at: https://paginas.fe.up.pt/~acbrito/laudon/ch7/chpt7-2main.htm#:~:text=A%20DBMS%20can%20reduce%20data,it%20can%20help%20control%20redundancy.
[Accessed 25 March 2021].

SQLite Tutorial, 2021. *SQLite Python.* [Online]
Available at: https://www.sqlitetutorial.net/sqlite-python/
[Accessed 27 March 2021].

Teeling, M., 2012. *What is Data Integrity? Learn How to Ensure Database Data Integrity via Checks, Tests, & Best Practices.* [Online]
Available at: https://www.veracode.com/blog/2012/05/what-is-data-integrity#:~:text=Data%20integrity%20is%20imposed%20within,should%20not%20accept%20alphabetic%20data.
[Accessed 25 March 2021].

The Regents of the University of California, n.d. *What Is a Data Dictionary?.* [Online]
Available at: http://library.ucmerced.edu/
[Accessed 25 March 2021].

TutorialCup, 2021. *File Proccessing System.* [Online]
Available at: https://www.tutorialcup.com/dbms/file-processing-system.htm

Tutorialink, n.d. *DBMS.* [Online]
Available at: https://tutorialink.com/dbms/advantage-and-disadvantages-of-file-oriented-system.dbms

WATT, A., 2014. *Chapter 3 Characteristics and Benefits of a Database.* [Online]
Available at: https://opentextbc.ca/dbdesign01/chapter/chapter-3-characteristics-and-benefits-of-a-database/
[Accessed 25 March 2021].

Watt, A., n.d. *Chapter 1 Before the Advent of Database Systems.* [Online]
Available at: https://opentextbc.ca/dbdesign01/chapter/chapter-1-before-the-advent-of-database-systems/