Title: **Database Management Group Assignment Part 1**

Intake code: **UCDF2309ICT(SE)**

Group Number: **Group 28**

Module Name: **Database Management**

Module Code: **022024-MHL**

Lecturer: **Mr. Muhammad Huzaifa Ismail**

Group Member	TP Number
Ng Yvonne	TP076390
Lum Han Xun	TP076160
Heng Xin Hui	TP077232
Connie Puang Pei Qi	TP078412

Table of Content

1.0 Database and Database Management System

1.1 Disadvantages of file-based system

File-based system, often refers as file system, is a software system used to store, organize, retrieve, and manage data from physical storage medium. For instance, hard disks, solid-state drive (SSD), flash drives and more. File-based system store various types of data, includes text, audio, video, or images, along with accompanying metadata about the files. The metadata could consist of file name, file size, last save date, location on a directory etc. (Sullivan & O'Brien, n. d.)

While the file-based system has been a traditional approach to data storage and management, they also come with several limitation. First and foremost, **data redundancy** might happen without being noticed. Data redundancy occurs when multiple copies of identical data is stored in different locations, such as different tables, databases or across various platforms. Using our assignment scenario as a case, data redundancy is likely to happen when the data of the quantity of ordered food is stored in multiple locations. Data redundancy slows down the access speed of databases especially during data retrieval. Database size will grow larger and larger when large amount of identical data occupies the database storage, leading to longer loading time and increase the potential of data corruption. (Gillis, n. d.)

Data redundancy results in **data inconsistency**. Data is considered inconsistent when multiple copies of identical data fail to align with each other. This may happen due to wrong data entry, multiple data sources, differences in formatting, issues with data synchronization and more. For instance, as the quantity of ordered food might fluctuate due to order cancellations, if in some system the quantity is being updated but the rest of the entries remain unaltered, wrong output will be returned when data is retrieved. Therefore, data inconsistency will lead to **poor data integrity**, thus causing error in analysis and confusing. (Educative, n. d.) Individual may face significant challenges to correctly analyse and identify data patterns, thus affect the decision-making process. There is also a wasted of time in identifying and rectifying data inconsistencies.

Last but not least, **data security** issue may arise when using file-based system. Since file-based system often have basic access control to the files and data stored, enforcement of access restriction is much difficult. Data can potentially be access by unauthorized parties and updated without detection. (OpenAI, 2023) Using our assignment scenario as an example, the manager should not have access to see or change the data of the member's personal information such as account password and contact number. Moreover, when files are shared, it is challenging to track the person who shares and changes made. This increases the risk of data breaches and unauthorized modification. (Educative, n. d.)

1.2 Advantages of Database and DBMS

A **database** is a structured set of information or data kept electronically in a computer system. It is arranged to be easily accessed and updated and is usually managed by a DBMS (OCI, n.d.). A **DBMS** refers to system software designed to create and manage databases, acting as a bridge between the database and users or applications which can provide a few advantages (Mullins, 2023). The primary advantage of a database and DBMS is to **provide better data security**. DBMS can improve data security through user authentication and user privileges (Pedamkar, 2023). This helps to ensure the confidentiality of data as only authorized users can access it, thus effectively preventing security issues such as theft and manipulation (Aggarwal, 2023). For example, when students register as members of the online system, their **member IDs** and **passwords** are securely stored in the database. When members wish to log in, the DBMS verifies their authentication details before allowing access to their account status. To ensure the confidentiality of members' personal information, only authorized users such as staff will have access to their accounts.

Besides, the advantage of DBMS is its ability to **improve data integrity and consistency**. Essentially, the consistency of data in a DBMS can be maintained by enforcing rules and constraints on the data to ensure data accuracy and authenticity. This includes actions such as avoiding duplicate entries and ensuring that each record has a distinct identifier to reduce the risk of data errors (ScaleGrid, 2024). For instance, the DBMS ensures real-time updates on payment status. When a member completes payment, the system immediately updates the **payment status** to "Success" or "Failed" to ensure accuracy for members and cafe managers.

Also, the DBMS can maintain consistency in restaurant location data and food menu names. For example, if the cafe decides to change its **restaurant location** or update its **food menu names**, the DBMS can ensure that these changes are reflected consistently throughout the system. This can help to avoid inconsistencies such as outdated food menu names or incorrect restaurant addresses, thereby improving the overall user experience for members.

Furthermore, a DBMS can also **ensure better recovery and backup**. Users have no need to do backups manually as this is handled by the DBMS. Fundamentally, a robust framework for data backup and recovery is offered by a DBMS, ensuring the recovery of database data after a failure by methods including backup, replication, and logging. (Aggarwal, 2023). This not only saves time and money, but also reduces the risk of data loss due to human error and minimizes disruption (E,2021). For example, essential information such as **member and worker IDs** can remain protected and accessible even in the event of a system failure or technical issues.

Moreover, DBMS is capable of **decreasing data redundancy** by minimizing the occurrence of isolated files that repetitively store the same data (Ninth Edition, n.d.). In a DBMS managed database, all data is stored in a database. Changes made to the data are promptly updated in the database, thereby preventing duplication of data. For example, when a member submits feedback on a food item, the DBMS will update the **feedback ID** in the database to prevent duplicate entries and maintain data integrity. This ensures that data redundancy is significantly eliminated as data is stored in a structured way (Tutorialspoint, n.d.). Additionally, this structured approach ensures that critical information such as **order IDs** and **quantities of food** is stored efficiently without unnecessary duplication.

In addition, DBMS can help to **improve better decision making**. Through well-managed data and enhanced data accessibility, DBMS ensures that users have access to accurate and relevant data (Aggarwal, 2023). As customer feedback, which is represented by **feedback IDs** and **customer ratings**, plays a crucial role in guiding business decisions within the cafe, the DBMS can ensure that users such as managers have easy access to the data so that cafe management can quickly analyze customer preferences and identify trends in order to meet customer preferences effectively. For example, if a certain food receives many low ratings, the cafe management can take measures to improve its quality and therefore enhance customer satisfaction and loyalty. Additionally, a DBMS provides a framework to facilitate data quality initiatives and enable organizations to implement better data management techniques. Higher quality information can be produced through better data management techniques, which in turn increases the validity and timeliness of data used for decision-making (Gorai, n.d.).

1.3 Functions of DBMS

On the other hand, DBMS also consists of many functions. First of all, DBMS plays an important role in **data storage management**. DBMS stores and organizes metadata and efficiently processing various data formats. It helps users by structuring complex data sets so that it can be easily accessed and manipulated. Effective data storage management is critical to optimizing database performance, thereby increasing the speed and efficiency of data access. For example, when a member places an order, the DBMS quickly retrieves the relevant information, such as **food prices** and **order IDs** to ensure a seamless user experience. Overall, a DBMS ensures that data is stored and organized to facilitate users (Javatpoint, n.d.).

Apart from that, DBMS functions include **data dictionary management**. DBMS keeps data metadata and element definitions in a data dictionary, simplifying coding for developers by managing the relationships between data structures within it (Javatpoint, n.d.). For example, the data dictionary specifies how **ApFood wallet IDs** are associated with **member IDs**, thereby facilitating seamless transactions within the online food ordering system. In addition, the data dictionary also automatically records any changes made to the database structure (My Reading Room, n.d.). For example, when a member adds funds to their ApFood wallet, the DBMS records the new balance. Likewise, deductions from the wallet for order payments or refunds are also recorded.

Furthermore, the function of DBMS is **multiuser access control**. DBMS follows ACID properties to maintain database consistency, especially when multiple users access the database simultaneously. (Javatpoint, n.d.). For example, when multiple members place orders at the same time or when cafe staff update inventory information simultaneously, the DBMS ensures that each transaction is processed atomically, therefore maintaining data consistency throughout the system.

Lastly, DBMS can perform **data transformation and presentation tasks**, converting input data into the necessary data structures and storing it in the specified format. There is no need to be concerned with the logical and physical representation of the data (Javatpoint, n.d.). For example, when a member submits an order through the online food ordering system, the DBMS converts the order details including **food items**, **quantities**, and **delivery status** into a structured format suitable for storage and processing.

2.0 Business Rules & Normalization

2.1 Business Rules

Business rules is a constraint statement to a certain aspect of database. For example, business rules allow us to figure out relationships between the entities in a database clearly to ensure each data is useful. An organization usually create business rules during the database-design process to know the requirements of a business, what data to store, and the relationships between data (eTutorials.org, n.d.).

1. Each member can own one shopping cart only. Each shopping cart is owned by one member.
2. Each member can give zero or more feedback. Each feedback is given by only one member.
3. Each member can have zero or many food purchase histories. Each food purchase history is owned by one member only.
4. Each member can only own an APfood wallet. Each APfood wallet belongs to only one member.
5. Each member can earn zero or more reward details. Each reward detail is earned by one member.
6. Each member can place zero or many orders. Each order can be placed by one member.
7. Each food can receive zero or more feedback. Each feedback can only receive by one food.
8. Each food is offered by one restaurant only. Each restaurant can offer one or more food.

9. Each payment detail can validate one or many orders. Each order is validated by one payment detail only.
10. Each order can include one or more food. Each food can be included in zero or many orders.
11. Each shopping cart encompasses zero or many orders. Each order can be encompassed by a shopping cart.
12. Each shopping cart can attach zero or many payment details. Each payment detail is attached to only one shopping cart.
13. Each worker can deliver zero or many orders. Each order can be delivered by one worker.
14. Each chef can prepare zero or many orders. Each order can be prepared by one or many chefs.
15. Each chef is supervised by one manager. Each manager can supervise one or many chefs.
16. Each restaurant can employ one or many chefs. Each chef works in one restaurant only.
17. Each payment detail can be viewed by one member only. Each member can view zero or many payment details.
18. Each payment detail is matched to a food purchase history. Each food purchase history is matched to one and only one payment detail.
19. Each payment detail can be associated with one TNG receipt only. Each TNG receipt is associated with one payment detail.
20. Each payment detail can be associated with one online banking receipt only. Each online banking receipt is associated with one payment detail.
21. Each payment detail can be associated with one APfood wallet receipt only. Each APfood wallet receipt is associated with one payment detail.
22. Each payment detail can link to one reward detail. Each reward detail can be linked to one payment detail.
23. A manager can compile zero or many orders. Each order can be compiled by a manager.
24. Each APfood wallet can store zero or many payment details. Each payment detail can be stored in an APfood wallet only.
25. Each APfood wallet can contain zero or many reward details. Each reward detail can contain in an APfood wallet.
26. Each food purchase history can track one and only one order. Each order can be tracked by one food purchase history.
27. Each reload history can only be recorded in one APfood wallet. Each APfood wallet can record zero or many reload histories.

28. Each TNG receipt can only be received by one member. Each member can receive zero or many TNG receipts.
29. Each online banking receipt can only be received by one member. Each member can receive zero or many online banking receipts.
30. Each APfood wallet receipt can only be received by one member. Each member can receive zero or many APfood wallet receipts.

2.2 Normalization

Normalization is defined as the structured process of creating a database model by categorizing its attributes. Normalization is critical for database management because it ensures data consistency and efficiency while minimizing data redundancy (GeeksForGeeks, 2024). Data redundancy occurs when the same data appears in many databases. As a result, the database will grow in size, causing plenty of issues in the future.

Apart from reducing data redundancy, there are a plethora of reasons for the importance of normalization. For instance, data management is made easier by breaking a complex data structure into manageable tables for retrieval purposes. Furthermore, normalization guarantees that data structures are flexible, allowing them to adjust to changing business requirements. As organizations grow, they can make changes to their databases without harming data accuracy. As a result, normalization benefits businesses seeking ordered data structures. (Rehan, 2023)

UNF

Normalization involves several phases, which are commonly referred to as normal forms. Normal forms include Unnormalized Form (UNF), First Normal Form (1NF), Second Normal Form (2NF), and Third Normal Form (3NF). UNF is a database's raw and unnormalized data, which means it contains redundancy and may produce unexpected findings (Romani, 2023). Diagram 2.2.1 shows the UNF table in our database.

FoodOrderingSystem (feedbackID, description, rating, memberID, mName, mUsername, mPassword, mGender, mRole, mContactNumber, foodID, fName, price, restaurantID, rName, rContactNumber, location, halalLicense, cleanlinessLicense)

FoodOrdering System

feedbackID	description	rating	memberID	mName	mUsername	mPassword	mGender	mRole	mContactNumber	foodID	fName	price	restaurantID	restaurantName	rContactNumber	location	halalLicense	cleanlinessLicense
E01	The taste is niceeee.	4.5	M01	Harry	harry123	23h483ej	Male	Student	012-334 5637	F05	Iced lemon tea	RM3.50	RT02	Fizz and Sip Lounge	03-5930274	east	HC63826	A-
E02	The meat is undercooked!!!!!	2.0	M01	Harry	harry123	23h483ej	Male	Student	012-334 5637	F01	Cheeseburger	RM4.50	RT01	Savor Junction	03-2341722	south	HC28439	A
E03	I won't be ordering this again.	1.5	M02	Matthew	5_matt_5	dj37dj9d	Male	Student	017-805 1823	F15	Tiramisu	RM4.50	RT03	Sweet Bliss Delight	03-2374920	west	HC90631	A+
E04	Definitely a must-try for pizza lovers!	4.9	M02	Matthew	5_matt_5	dj37dj9d	Male	Student	017-805 1823	F11	Pepperoni Pizza	RM10.00	RT01	Savor Junction	03-2341722	south	HC28439	A

Diagram 2.2.1 UNF Table

1NF

From UNF to 1NF, data inconsistency is reduced by deleting repetitive and repeating data. Each property must contain a single appropriate value, not multiple values or nulls. Next, primary keys are identified to uniquely represent attribute values. Finally, the dependencies are determined. There are two sorts of dependencies: partial and transitive. A partial dependency is an attribute that depends on one or more primary keys, whereas a transitive dependency is a non-primary key attribute that relies on another non-primary key attribute. Partial dependency creates a direct relationship between characteristics, whereas transitive dependency creates an indirect relationship between attributes (Grainger, 2021). The primary keys, partial, and transitive dependencies for our database are listed below.

Primary Keys:

- feedbackID
- memberID
- foodID
- restaurantID

Partial Dependencies:

- feedbackID, memberID, foodID -> description, rating
- memberID -> mName, mUsername, mPassword, mGender, mRole, mContactNumber
- restaurantID -> rName, rContactNumber, location, halalLicense, cleanlinessLicense, foodID, fName, price

Transitive Dependencies:

- restaurantID -> foodID -> fName

2NF

From 1NF to 2NF, various rows of tables containing data sets are removed and replaced with new tables. Subsequently, linkages between the new tables are established. To generate a new table that is 2NF, 1NF must be satisfied and partial dependencies are removed before being migrated to a new table (Morris, 2022). Diagram 2.2.2 represents our group's 2NF table.

Existing table:

- Feedback (feedbackID, memberID, foodID, description, rating)
- Member (memberID, mName, mPassword, mGender, mRole, mContactNumber)
- Restaurant (restaurantID, rName, rContactNumber, location, halalLicense, cleanlinessLicense, foodID, fName, price)

Feedback

feedbackID	description	rating	memberID	foodID
E01	The taste is niceeee.	4.5	M01	F05
E02	The meat is undercooked!!!!!	2.0	M01	F01
E03	I won't be ordering this again.	1.5	M02	F15
E04	Definitely a must-try for pizza lovers!	4.9	M02	F11

Member

memberID	mName	mUsername	mPassword	mGender	mRole	mContactNumber
M01	Harry	harry123	23h483ej	Male	Student	012-334 5637
M02	Matthew	5_matt_5	dj37dj9d	Male	Student	017-805 1823
M03	Evelyn	evelynnn	4372hd82	Female	Staff	012-444 1016
M04	Malfoy	888malfoy888	dihd8999	Male	Staff	018-845 1584

Restaurant

restaurantID	rName	rContactNumber	location	halalLicense	cleanlinessLicense	foodID	fName	fPrice
RT01	Savor Junction	03-2341722	south	HC28439	A	F01	Cheeseburger	RM4.50
RT02	Fizz and Sip Lounge	03-5930274	east	HC63826	A-	F02	Molten lava cake	RM7.20
RT03	Sweet Bliss Delight	03-2374920	west	HC90631	A+	F03	Spaghetti	RM8.50

Diagram 2.2.2 2NF table

3NF

From 2NF to 3NF, no non-primary key will develop transitive dependency on a primary key. Hence, the database must have no transitive dependency, which means that no attribute can affect the output of another. To establish a 3NF table, 1NF and 2NF must be fulfilled, with no transitive dependency for non-primary keys (Morris, 2022). Diagram 2.2.3 indicates our 3NF table.

Existing table:

- Feedback (feedbackID, memberID, foodID, description, rating)
- Member (memberID, mName, mPassword, mGender, mRole, mContactNumber)
- Restaurant (restaurantID, rName, rContactNumber, location, halalLicense, cleanlinessLicense, FoodID)
- FoodMenu (foodID, fName, price)

Feedback

feedbackID	description	rating	memberID	foodID
E01	The taste is niceeee.	4.5	M01	F05
E02	The meat is undercooked!!!!!	2.0	M01	F01
E03	I won't be ordering this again.	1.5	M02	F15
E04	Definitely a must-try for pizza lovers!	4.9	M02	F11

Member

memberID	mName	mUsername	mPassword	mGender	mRole	mContactNumber
M01	Harry	harry123	23h483ej	Male	Student	012-334 5637
M02	Matthew	5_matt_5	dj37dj9d	Male	Student	017-805 1823
M03	Evelyn	evelynnn	4372hd82	Female	Staff	012-444 1016
M04	Malfoy	888malfoy888	dihd8999	Male	Staff	018-845 1584

Restaurant

restaurantID	rName	rContactNumber	location	halalLicense	cleanlinessLicense	foodID
RT01	Savor Junction	03-2341722	south	HC28439	A	F01
RT02	Fizz and Sip Lounge	03-5930274	east	HC63826	A-	F02
RT03	Sweet Bliss Delight	03-2374920	west	HC90631	A+	F03

FoodMenu

foodID	fName	fPrice
F01	Cheeseburger	RM4.50
F02	Molten lava cake	RM7.20
F03	Spaghetti	RM8.50
F04	Veggie wrap	RM6.00

Diagram 2.2.3 3NF table

In conclusion, companies that manage and depend on immense quantities of data are required to be familiar with normalization because it effectively organizes data structures. This reduces database costs by reducing storage and size. Aside from that, database security improves when data is properly sorted and arranged. Without normalization, more effort is required locating the proper data since even little changes, such as changing an attribute, generate problems in other attributes in the database. Therefore, enterprises will receive several benefits and advantages from normalization in the future (Morris, 2022).

3.0 Entity Relationship Diagram

Diagram 3.0.1 below is the Entity Relationship Diagram of our case study about APU café food ordering system. It clearly shows how the entities and attributes connected to each other.

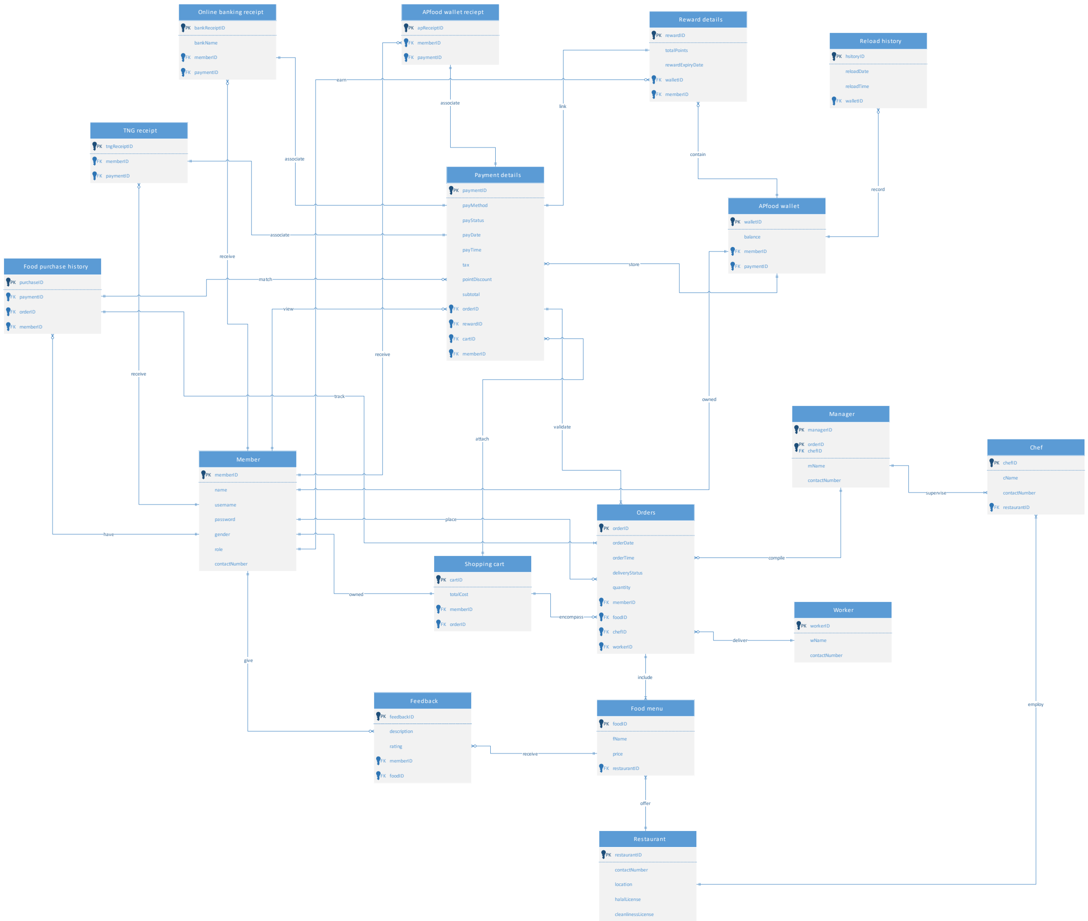

Diagram 3.0.1 Entity Relationship Diagram of APU café food ordering system

4.0 References

Aggarwal, C. (2023). *What are the Advantages of DBMS?* Shiksha Online. https://www.shiksha.com/online-courses/articles/what-are-the-advantages-of-dbms/

E. (2021). *13 Advantages of Database Management System in Email Marketing*. Enginemailer.https://www.enginemailer.com/blog/advantages-database-management-system

Educative. (n. d.). *Disadvantage of file-based system*. https://www.educative.io/courses/database-design-fundamentals/disadvantages-of-file-based-system

eTutorials.org. (n.d.) *What are business rules?* https://etutorials.org/SQL/Database+design+for+mere+mortals/Part+I+The+Design+Process/Chapter+11.+Business+Rules/What+Are+Business+Rules/

GeeksForGeeks. (2024). *Introduction of Database Normalization*. https://www.geeksforgeeks.org/introduction-of-database-normalization/

Gillis, A. S. (n. d.). *Redundant*. TechTarget. https://www.techtarget.com/searchstorage/definition/redundant

Gorai, S. (n.d.). *DBMS advantages and Disadvantages*. Scribd. https://www.scribd.com/document/383038404/DBMS-advantages-and-Disadvantages

Grainger, C. (2021). *Database Normalization — How it Works*. Medium. https://chasegrainger.medium.com/database-normalization-how-it-works-51f79d802042

Javatpoint. (n.d.). *Functions of DBMS*. https://www.javatpoint.com/functions-of-dbms

Morris, S. (2022). *Data Normalization: Definition, Importance, and Advantages*. Coresignal. https://coresignal.com/blog/data-normalization/

Mullins, C. S. (2023). *Database Management System (DBMS)*. TechTarget. https://www.techtarget.com/searchdatamanagement/definition/database-management-system

Ninth Edition. (n.d.). *Management Information Systems*. https://paginas.fe.up.pt/~acbrito/laudon/ch7/chpt7-2main.htm

OpenAI. (2023). *ChatGPT (Mar 14 version) [Large language model]*. https://chat.openai.com/chat

Oracle. (n.d.). *What is a Database?* https://www.oracle.com/my/database/what-is-database/

Pedamkar, P. (2023). *Advantages of DBMS*. EDUCBA. https://www.educba.com/advantages-of-dbms/

Rehan, M. (2023). *What is Data Normalization and Why is it Important?* Apimio. https://apimio.com/importance-of-data-normalization/

Romani, E. (2023). *What Is Database Normalization?* Built In. https://builtin.com/data-science/database-normalization

ScaleGrid. (2024). *Advantages Of DBMS In Data Management*. https://scalegrid.io/blog/advantages-of-dbms/

Sullivan, E. & O'Brien, E. (n. d.). *File system*. TechTarget. https://www.techtarget.com/searchstorage/definition/file-system

T. (n.d.). *Functions of Database Management System (DBMS)*. Myreadingroom. https://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/465-functions-of-dbms.html

Tutorialspoint. (n.d.). *Advantages of Database Management System*.

https://www.tutorialspoint.com/Advantages-of-Database-Management-System

AICT005-4-1 Database Systems – Workload Matrix

Part	Component	Student 1 Name: Ng Yvonne	Student 2 Name: Lum Han Xun	Student 3 Name: Heng Xin Hui	Student 4 Name: Connie Puang Pei Qi	Total
1	a) Database and Database Management System		**50%**	**50%**		**100%**
1	b) Business Rules & Normalization	**45%**	**10%**		**45%**	**100%**
1	c) Entity Relationship Diagram	**100%**				**100%**