D.B.M.S

What is DBMS?

Database Management system also known as DBMS is a software that is created to manage, specify, retrieve and store data in a database.

Components of DBMS

- Hardware
- Data
- People
- Procedures
- software

Brief explanation and Examples of components of DBMS

- **Hardware** –"hardware" refers to the actual components that make up a computer. It also includes different storage devices, such as hard drives, and input and output devices, such as printers. For DBMS Servers must run the DBMS software, and there must be a location to store the data. Even cloud services need hardware in today's increasingly cloud-enabled era, even if those servers aren't actual boxes in a server room down the hall.
- **Data** – The collection of information kept in the database is referred to as data. Considering that data are the foundation from which information is created. Structured, non-structural, and logical data are all types of data that are kept in databases. A DBMS is useless if you don't have any data to store in it. A person name is data.
- **People** – A database is used by many various types of people, including administrators who are responsible for the software's flawless operation, application developers who use the data in their products, and end users who actually generate and use the data.
- **Procedures** – Procedures are broad guidelines for utilizing a database management system. This comprises steps for installing a DBMS, logging in and out of DBMS software, managing databases, creating backups, and reports, among other things.
- **Software** – The software makes up the majority of a database management system. Software is a group of programs or a set of programs that provide commands to a computer. An example of software is MySQL.

<u>Question 2</u>

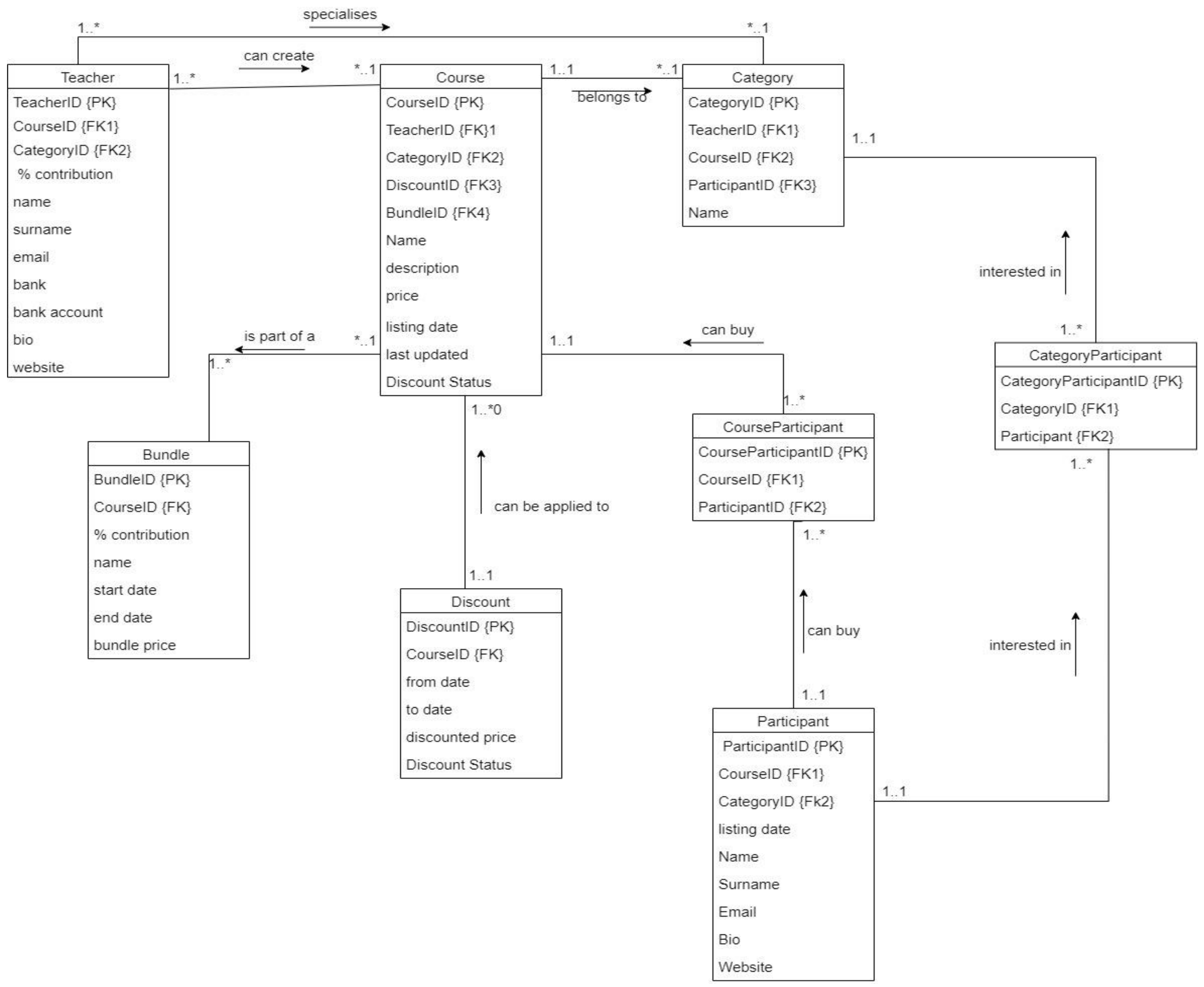
Q2.1

With the data about Courses I would advise using a relational database, which is a type of database that stores and allows access to data that is related to one another. Afterwards, this data can be built up as a collection of tables with columns and rows. Tables can store data about the objects or entities that will be used as database representations. Within the database, users can create links between various kinds of data, which they can then use to manage and organize relevant data. Relational databases use SQL, also known as Structured Query Language, to carry out queries and manage data, and all of this appears to be achievable given the data that has been mentioned.

Q2.2

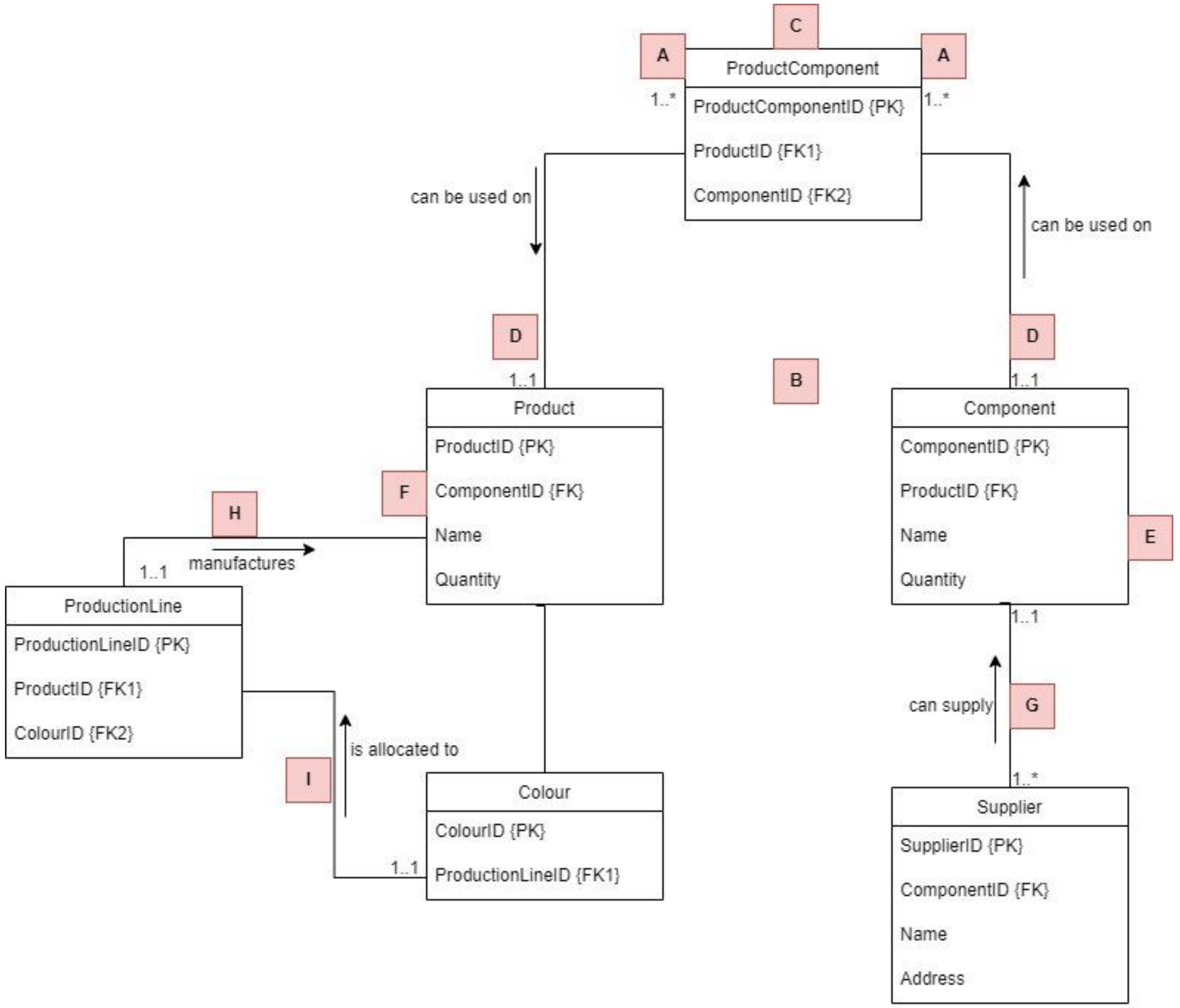
I would recommend a NoSQL database NoSQL is an alternative to the relational table-based schema used in SQL databases. Instead, NoSQL databases handle and store data using a range of data models, including document-oriented, key-value, graph, and column family models. NoSQL databases are perfect for large volume applications since they are built to handle massive volumes of unstructured or semi-structured data. They are also very scalable and able to deal with lots of simultaneous users and requests. Web applications, social networking platforms, and other data-intensive applications frequently use NoSQL databases. No SQL databases are flexible, allowing the structure of the data to be modified without affecting the entire database. They are designed to manage unstructured and semi-structured data, making them perfect for storing multimedia files. This makes it simpler to manage various multimedia files with diverse structures. The data mentioned is unstructured data that requires large storage and NoSQL enables faster, more agile storage and processing for all users making it more suitable to take on the data the mentioned.

<u>Question 3</u>


specialises
1..*
*..1
can create
1..*
*..1
1..1
*..1
belongs to
Teacher
TeacherID {PK}
CourseID {FK1}
CategoryID {FK2}
% contribution
name
surname
email
bank
bank account
bio
website
Course
CourseID {PK}
TeacherID {FK}1
CategoryID {FK2}
DiscountID {FK3}
BundleID {FK4}
Name
description
price
listing date
last updated
Discount Status
Category
CategoryID {PK}
TeacherID {FK1}
CourseID {FK2}
ParticipantID {FK3}
Name
1..1
interested in
1..*
CategoryParticipant
CategoryParticipantID {PK}
CategoryID {FK1}
Participant {FK2}
1..*
is part of a
*..1
1..*
1..1
can buy
1..*
1..*0
Bundle
BundleID {PK}
CourseID {FK}
% contribution
name
start date
end date
bundle price
can be applied to
1..1
Discount
DiscountID {PK}
CourseID {FK}
from date
to date
discounted price
Discount Status
CourseParticipant
CourseParticipantID {PK}
CourseID {FK1}
ParticipantID {FK2}
1..*
can buy
1..1
interested in
Participant
ParticipantID {PK}
CourseID {FK1}
CategoryID {Fk2}
listing date
Name
Surname
Email
Bio
Website
1..1

Question 4

I would change the name Item and use the word Product as in the business rules theproducts are called product and not item this would reduce confusion for both the users of the database as well as for Thato. Then I would set my primary keys for all the tables. In business rule number one it says that “a component can be used in multiple different products,” and this is a one-to-many relationship meaning one component on many products. But on the diagram Thato made a one-to-one relationship making the first business rule incorrect. Due to there being 2 one to many relationships between product and component that means there is an associative table that has to be created. This associative table will have componentID and ProductID as its foreign keys. The relationship between product and component will be removed as indicated by the letter “B”, and the two tables will have their relationship through the new table ProductComponent shown below letter “C”. the relationship will be a one to one shown by the letter “D” and one to many relationship the many being shown on the associative table ProductComponent, shown by letter “A”. To satisfy business rule 2 I would add “quantity” as an attribute in the table Component shown by letter “E” and add ComponentID as a foreign key in the table Product, shown by letter “F”. The “name” of each component and Product is already saved as attributes in the 2 tables, thus satisfying business rule number 3.To satisfy business rule number 4 we add a link between component and supplier then we add the relationships because each component is bought from a specific supplier that is a one to one relationship, and a supplier can supply many different components this is a one to many relationship, this link is shown by the letter “G”. I would also add componentID as a foreign key in the supplier table. To satisfy business rule number 5 I would add “name” and “address” as attributes of table supplier. To satisfy business rule number 6 we add a table called ProductionLine and add productID as a foreign key, then we add the link and relationship between the 2 tables shown by letter “H” which will be a one to one relationship since each production line manufactures exactly one type of product at a time. to satisfy business rule number 7 we create a link between the table production line and table color shown by letter “I” then add a relationship between the two tables which is a one to one relationship because one color is allocated to one production line, then we add our foreign keys to the two tables so that the data in the tables can be found using their foreign keys. We can add comments and arrows to make the diagram readable and easier to understand. I would remove the link between table supplier and table product because there is no business rule stating a relationship between the 2 making it a useless link, the supplier already supplies the component which is already linked to the product. Below shows how the diagram would look like after the business rules have been satisfied and the mistakes removed.



C
A
ProductComponent
A
1..*
ProductComponentID {PK}
1..*
ProductID {FK1}
ComponentID {FK2}
can be used on
can be used on
D
D
B
1..1
1..1
Product
ProductID {PK}
ComponentID {FK}
Name
Quantity
F
Component
ComponentID {PK}
ProductID {FK}
Name
Quantity
E
H
manufactures
1..1
ProductionLine
ProductionLineID {PK}
ProductID {FK1}
ColourID {FK2}
is allocated to
I
Colour
ColourID {PK}
1..1
ProductionLineID {FK1}
1..1
can supply
G
1..*
Supplier
SupplierID {PK}
ComponentID {FK}
Name
Address

References

education, I. I. (2012). *DATABASES module manuel.*

What are the Components of DBMS (Database Management System)? (2022, September 6). Retrieved from InterviewBit: https://www.interviewbit.com/blog/components-of-dbms/

What is a Relational Database (RDBMS)? (n.d.). Retrieved from OCI: https://www.oracle.com/database/what-is-a-relational-database/

Bibliography

education, I. I. (2012). *DATABASES module manuel.*

What are the Components of DBMS (Database Management System)? (2022, September 6). Retrieved from InterviewBit: https://www.interviewbit.com/blog/components-of-dbms/

What is a Relational Database (RDBMS)? (n.d.). Retrieved from OCI: https://www.oracle.com/database/what-is-a-relational-database/

Works Cited

education, I. I. (2012). *DATABASES module manuel.*

What are the Components of DBMS (Database Management System)? (2022, September 6). Retrieved from InterviewBit: https://www.interviewbit.com/blog/components-of-dbms/

What is a Relational Database (RDBMS)? (n.d.). Retrieved from OCI: https://www.oracle.com/database/what-is-a-relational-database/