Introduction

Based on our understanding to the quote we have been provided in the homework, we found out that we need to have the following attributes:

- UserID (every student/teacher needs a unique a user ID)
- UserName (every user has a name)
- UserType(since we have different types of users i.e., students/teachers)
- Phisical_ID (every Book has unique ID even if books with the name and Auther but different copy number)
- Title(the name of each book)
- Author(the Author of a specific book)
- ISBN(International Standard Book Number)
- Edition(what is the edittion of the book, example: 1^{st},2^{nd},3^{rd},etc...)
- Gener(which gener the book has, example: action,drama,scifi, etc...)
- Language(what language the book is written in)
- BorrowingDate(The date when a specific book got borrowed from the library)
- ReturnDate(The date when a specific book got returned to the library)
- Duration (how many days the book was with the user until returned to the library)
- FineAmount(the amount of money per day which a user needs to pay if he/she returned a book late)

According to this information we were able to find the following FDs:

- UserID => UserName, UserType, Adress, Email
- Book_ID => Title, Author, ISBN, Edition, Gener, Language
- BorrowingID => Book_ID,BorrowingDate, ReturnDate, UserID

By applying BCNF algorithm on these FDs we got the following relations:

- R1(UserID, UserName, UserType, Email, Adress)
- R2(BookID, BookName, AuthurName)
- R3(BorrowingID, BookID, BorrowingDate, ReturnDate, UserID)
- R4(BorrowingID, FineAmount)

Answer for question number 1

First realtion will be user relation descibed as following:

UserID	UserName	UserType	Adress	Email
1234	Paula Hanna	Student	Somestreet 11	Something71@yahoo.com
5678	Dena Hussain	Teacher	Somestreet 45	Something72@gmail.com
0023	Raef Bechara	Student	Somestreet 11	Something7@hotmail.com

User(UserID : Integer, UserName: String, UserType: String, Adress: string, Email: String)

Explanation:

- Every user need to have a unique user ID in order to be differentiated from other users and that's why we chose to have UserID attribute in this relation. Since user ID is unique and can't be duplicated, we chose to have it as a primary key in this relation

- It's logical and important to have the most basic information about each user which is his/her name. Based on that we decided to have UserName attribute which hold each user's name.

- Since we have different types of users, we considered it as a necessity to have an attribute which states what is the type of user I.e., Student or Teacher. Based on that we chose to have UserType attribute.

- There are few reasons why having an Adress attribute is useful, for example if the student/teacher were to be fined, the bill can be sent to his/her address.

- Finally, Email address is quite good optimization and also can be useful in many ways, for instance fines bills can be sent via email in addition to sending them to the address. Another example of how email could be useful is when the library wants to inform student/teacher that the book he/she was waiting for is available

now.
Observation: even though email address is unique for each user, yet it couldn't be used as a primary key, because the user might need to change it later or stop using the email he/she registered with.

Second relation will be book relation:

Physical_ID	Title	Author	ISBN	Edition	Gener	Language
0023123	Harry Potter and the Prisoner of Azkaban	Joanne Rowling	Some serial number	1st	Action	English
1235666	Lord of The Rings	John Tolkien	Some serial number	3rd	Action	English
0023777	Harry Potter and the Prisoner of Azkaban	Joanne Rowling	Some serial number	5th	Action	English

Book(Physical_ID : Integer, Author: String, ISBN: String, Edition: String, Language: String)

Explanation:

- Every book must have a book ID in order to be unique from the other books. Two copies of the same book will have different IDs. That's why we chose to include Physical_ID attribute which hold the Book ID that makes it unique from all the other books and that's is why we are using it as a primary key in this relation.

- Title, Author, Edition, Gener and language for each book need to be saved in order to identify every book. It's also useful to have such information that can help both teachers and students to categorize books and makes it much easier to have filters on them which will make searching for as specific book much easier for users.

- Finally, we have the ISBN for each book, which stands for International Standard Book Number which is a unique number for each book and can proive informtion such as were the book was printed etc....

Observation: Although ISBN is a unique number for each book but it can happen that two copies of the same book have the same ISBN and that's why we didn't choose it to be our primary key for this relation.

Third relation will be Borrowed Books realtion:

BorrowingID	Physical_ID	UserID	BorrowingDate	ReturnDate
01230333	0002222	1234	2021-09-24	2021-09-30
01324323	0001111	5678	2021-09-20	null

BorrowedBook(BorrowingID: Integer, Physical_ID: Integer, UserID: Integer, BorrowingDate: Date, ReturnDate: Date)

Explanation:

- Firstly, we needed to find something than can make a borrowed bool unique in order to have it as a primary key. Knowing that a book can be loaned to the same user more than one time which means that neither the Physical_ID of the book nor the UserID be unique in this relation. That's why we chose to have a BorrowingID as the unique identifier for this relation and that's we have it as a primary key.

- Having the physical_ID for books as an attribute seemed naturally logical since it will guide us to know which book exactly is borrowed. Also, it's a great link between the Book and the BorrowedBook relation. That's why it seemed useful to have Physical_ID from the Book relation as a foreign key in the BorrowedBook relation.

- We needed to know to whom each book is borrowed. Despite that the UserID can link the BorrowedBook relation to the User relation which can provide us with the user information and most importantly the user type I.e. whether the user is teacher or a student since both have different limitations when it comes to the

number of books they can borrow and the number of days they can have the book. That's why we chose to have the UserID as a foreign key in the BorrowedBook relation

- Both BorrowingDate and ReturnDate determine when was a book borrowed and returned which is an important information that can help us to know if the user who borrowed the be to be fined or not. That's why we chose to have both attributes in this relation.

Observation: The ReturnDate attribute can be assigned to which will mean that the book is borrowed but has not returned yet.

Another forth relation is Fine Relation

BorrowingID	FineAmount
01230034	50
02134443	75.5

Fine(BorrowingID: Integer,FineAmount: Double)

Explanation:

-First and most important thing to explain is, why don't we just add FineAmount attribute to the BorrowedBook relation knowing that it is also dependent on the BorrowingID which is the primary key in the BorrowedBook relation. The answer for that would be that we don't need to have a lot of not needed null values since most of the book borrowed probably return in time and there is no fine on them.

- FineAmount is important to know how much is the fine.
-BorrowingID is important link between the Fine relation and the BorrowedBook realtion.

Answer for second Question:

Already answered as our third relation on the first question, but there it is again:

Third relation will be Borrowed Books realtion:

BorrowingID	Physical_ID	UserID	BorrowingDate	ReturnDate
01230333	0002222	1234	2021-09-24	2021-09-30
01324323	0001111	5678	2021-09-20	null

BorrowedBook(BorrowingID: Integer, Physical_ID: Integer, UserID: Integer, BorrowingDate: Date, ReturnDate: Date)

Explanation:

- Firstly, we needed to find something than can make a borrowed bool unique in order to have it as a primary key. Knowing that a book can be loaned to the same user more than one time which means that neither the Physical_ID of the book nor the UserID be unique in this relation. That's why we chose to have a BorrowingID as the unique identifier for this relation and that's we have it as a primary key.

- Having the physical_ID for books as an attribute seemed naturally logical since it will guide us to know which book exactly is borrowed. Also, it's a great link between the Book and the BorrowedBook relation. That's why it seemed useful to have Physical_ID from the Book relation as a foreign key in the BorrowedBook relation.

- We needed to know to whom each book is borrowed. Despite that the UserID can link the BorrowedBook relation to the User relation which can provide us with the user information and most importantly the user type I.e. whether the user is teacher or a student since both have different limitations when it comes to the number of books they can borrow and the number of days they can have the book. That's why we chose to have the UserID as a foreign key in the BorrowedBook relation

- Both BorrowingDate and ReturnDate determine when was a book borrowed and returned which is an important information that can help us to know if the user who borrowed the be to be fined or not. That's why we chose to have both attributes in this relation.

Observation: The ReturnDate attribute can be assigned to which will mean that the book is borrowed but has not returned yet.

Answer for third question

-Domain for BorrowingID: Integer since it can provide us with number in the range of 2 billion which is quite good range.

-Domain for Physical_ID: Integer since it can provide us with number in the range of 2 billion which is quite good range.

-Domain for ISBN: String also known as varchar. Since it is worldwide used as a String and can have dashes (-) which can't go with integers

-Edition,Gener and Language: String (varchar), since we user letters or mix of letters and numbers to describe them.