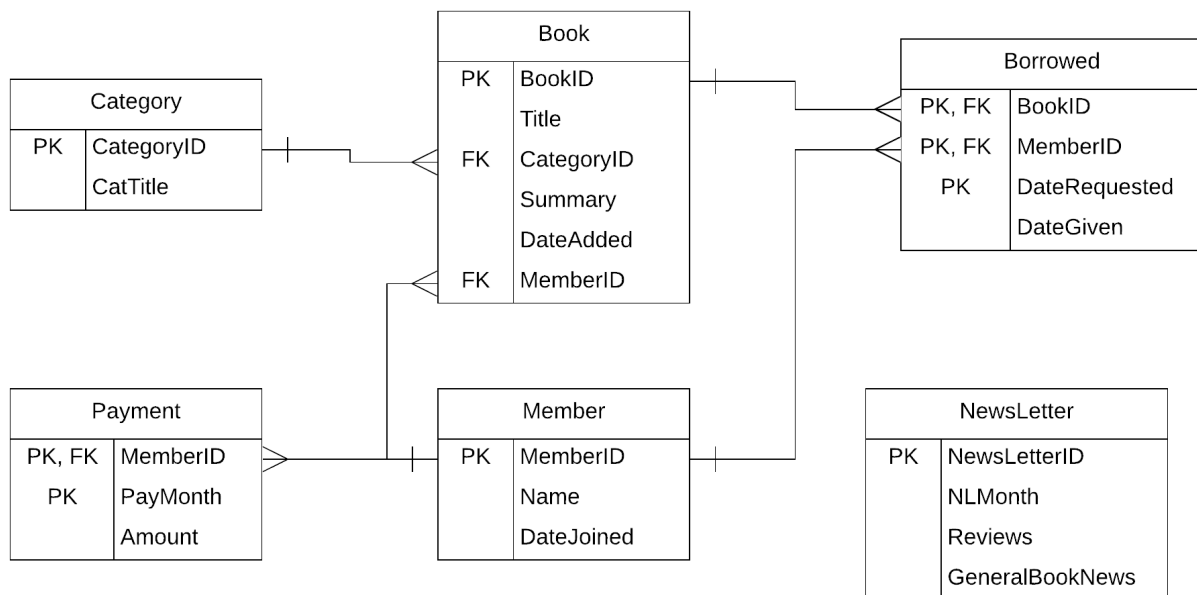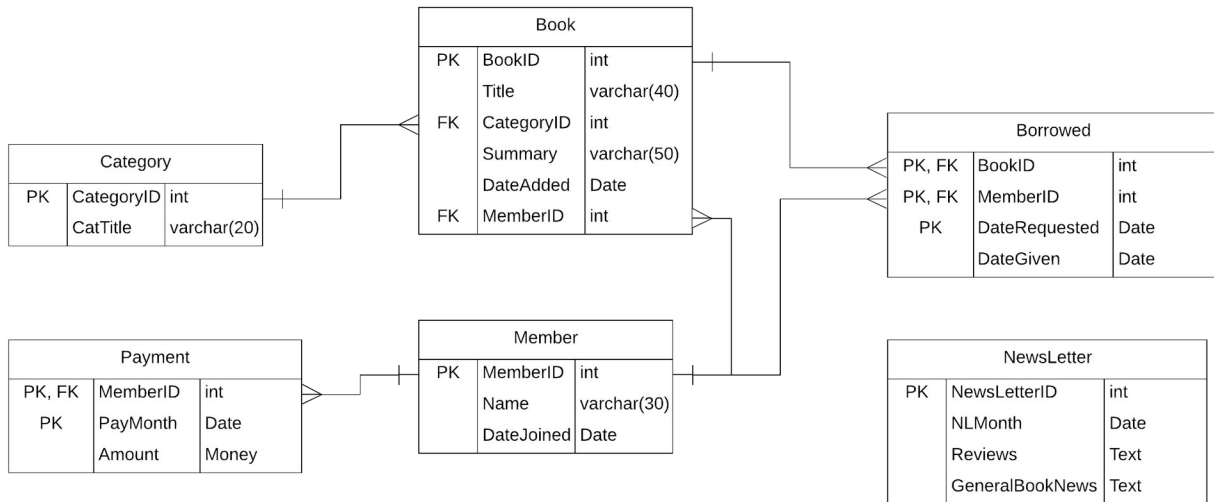## Some problems:

- DateAdded, DateRequested, DateGiven, DateJoined should be 'Date' data-type
- 'Title' of the 'Book' table should not be PK
- 'Summary' of the 'Book' table should be Text data-type
- 'Name' of the 'Member' table should not be PK
- The relationship between 'Book' and 'Member' should be refined 1 to many from Member to Book
- There should have a relationship between Member and Borrowed (1 to many)
- Phenomenon of data inconsistency: 'Member' field in Borowed and 'Name' field in Member
- Phenomenon of data redundancy: 'Book' table, 'Borrowed' table
- Lack of information about newsletter
- Lack of information about the monthly-payment

Note: You should find out more problems for this scenario
You should design your own ERD, together with your own explaination
My proposal is one of the solutions which maybe not the best one.

Book

PK	BookID
	Title
FK	CategoryID
	Summary
	DateAdded
FK	MemberID

Category

PK	CategoryID
	CatTitle

Borrowed

PK, FK	BookID
PK, FK	MemberID
PK	DateRequested
	DateGiven

Payment

PK, FK	MemberID
PK	PayMonth
	Amount

Member

PK	MemberID
	Name
	DateJoined

NewsLetter

PK	NewsLetterID
	NLMonth
	Reviews
	GeneralBookNews

Book

PK	BookID	int
	Title	varchar(40)
FK	CategoryID	int
	Summary	varchar(50)
	DateAdded	Date
FK	MemberID	int

Category

PK	CategoryID	int
	CatTitle	varchar(20)

Borrowed

PK, FK	BookID	int
PK, FK	MemberID	int
PK	DateRequested	Date
	DateGiven	Date

Payment

PK, FK	MemberID	int
PK	PayMonth	Date
	Amount	Money

Member

PK	MemberID	int
	Name	varchar(30)
	DateJoined	Date

NewsLetter

PK	NewsLetterID	int
	NLMonth	Date
	Reviews	Text
	GeneralBookNews	Text

Question 1: List member who have added some books this month:

Solution 1:

SELECT * FROM Member
WHERE MemberID IN(
        SELECT MemberID FROM Book
        WHERE MONTH(DateAdded) = MONTH(GETDATE()) AND YEAR(DateAdded) = YEAR(GETDATE())
)

$Member * (\Pi_{MemberID}(\sigma_{MONTH(DateAdded) = MONTH(GETDATE()) \wedge YEAR(DateAdded) = YEAR(GETDATE())} (Book))$

Solution 2:

SELECT Member.* FROM Member INNER JOIN Book ON Member.MemberID = Book.MemberID
WHERE MONTH(DateAdded) = MONTH(GETDATE()) AND YEAR(DateAdded) = YEAR(GETDATE())

$\Pi_{Member.*}(\sigma_{MONTH(DateAdded) = MONTH(GETDATE()) \wedge YEAR(DateAdded) = YEAR(GETDATE())} (Member*Book))$

Question 2: List members who have paid for newsletter this month:

Solution 1:

SELECT * FROM Member
WHERE MemberID IN(
SELECT MemberID FROM Payment
WHERE MONTH(PayMonth) = MONTH(GETDATE()) AND YEAR(PayMonth) = YEAR(GETDATE())
)

Member*($\Pi_{MemberID}$(σ MONTH(PayMonth) = MONTH(GETDATE()) ∧ YEAR(PayMonth) = YEAR(GETDATE()) (Payment))

Solution 2:

SELECT Member.* FROM Member INNER JOIN Payment ON Member.MemberID = Payment.MemberID
WHERE MONTH(PayMonth) = MONTH(GETDATE()) AND YEAR(PayMonth) = YEAR(GETDATE())

$\Pi_{Member.*}$(σ MONTH(PayMonth) = MONTH(GETDATE()) ∧ YEAR(PayMonth) = YEAR(GETDATE()) (Member*Payment))

Question 3: List member who have added some books OR have paid for newsletter this month:

Solution 1:

SELECT Member.* FROM Member INNER JOIN Payment ON Member.MemberID = Payment.MemberID
INNER JOIN Book ON Member.MemberID = Book.MemberID
WHERE (MONTH(PayMonth) = MONTH(GETDATE()) AND YEAR(PayMonth) = YEAR(GETDATE())) OR
(MONTH(DateAdded) = MONTH(GETDATE()) AND YEAR(DateAdded) = YEAR(GETDATE()))

$\Pi_{Member.*}$(σ(MONTH(PayMonth) = MONTH(GETDATE()) ∧ YEAR(PayMonth) = YEAR(GETDATE())) ∨ (MONTH(DateAdded) = MONTH(GETDATE()) AND YEAR(DateAdded) = YEAR(GETDATE())) (Member*Payment*Book))

Solution 2:

SELECT * FROM Member
WHERE MemberID IN(
SELECT MemberID FROM Book
WHERE MONTH(DateAdded) = MONTH(GETDATE()) AND YEAR(DateAdded) = YEAR(GETDATE())
)
UNION
SELECT * FROM Member
WHERE MemberID IN(
SELECT MemberID FROM Payment
WHERE MONTH(PayMonth) = MONTH(GETDATE()) AND YEAR(PayMonth) = YEAR(GETDATE())
)

Member*($\Pi_{MemberID}$(σ MONTH(DateAdded) = MONTH(GETDATE()) ∧ YEAR(DateAdded) = YEAR(GETDATE()) (Book)) U Member*($\Pi_{MemberID}$(σ MONTH(PayMonth) = MONTH(GETDATE()) ∧ YEAR(PayMonth) = YEAR(GETDATE()) (Payment))

Question 4: List members who have not added anybook this month:

Solution 1:

SELECT * FROM Member
WHERE MemberID NOT IN(
SELECT MemberID FROM Book
WHERE DATEPART(MONTH, DateAdded) = DATEPART(MONTH, GETDATE())
AND DATEPART(YEAR, DateAdded) = DATEPART(YEAR, GETDATE())
)

Member*($\Pi_{MemberID}$(Member) - $\Pi_{MemberID}$($\sigma_{\text{DATEPART(MONTH, DateAdded) = DATEPART(MONTH, GETDATE())} \wedge \text{DATEPART(YEAR, DateAdded) = DATEPART(YEAR, GETDATE())}}$ (Book))

Solution 2:

SELECT * FROM Member
EXCEPT
SELECT Member.*
FROM Member INNER JOIN Book ON Member.MemberID = Book.MemberID
WHERE DATEPART(MONTH, DateAdded) = DATEPART(MONTH, GETDATE())
AND DATEPART(YEAR, DateAdded) = DATEPART(YEAR, GETDATE())

Member - $\Pi_{Member.*}$(Member * ($\sigma_{\text{DATEPART(MONTH, DateAdded) = DATEPART(MONTH, GETDATE())} \wedge \text{DATEPART(YEAR, DateAdded) = DATEPART(YEAR, GETDATE())}}$ (Book)))

Notice that you have 2 hours for your exam, so I think you must make use of time to do well your exam paper.
I think you should explain a little more about your decision, such as why you design so, why you choose your answer and so on.
If you give the answers only, duration of 30 minutes is enough for you to complete the exam. So it is not good!