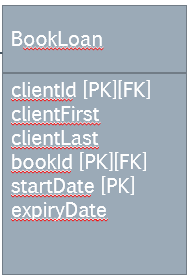
**Relational Modelling Quiz III**

**Question 1**

We are now dealing with an online book store where you can both buy and borrow books, electronic books and audiobooks. The following BookLoan relation is part of the database used to run this store. It is not the only table in the database, we choose not to show the others to keep things simple.

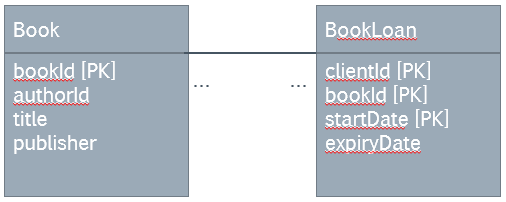


What, if anything, is a real problem here?

* The table breaks the relational principle of no redundancy, because the clientFirst and clientLast names describe the client, not the loan entity.

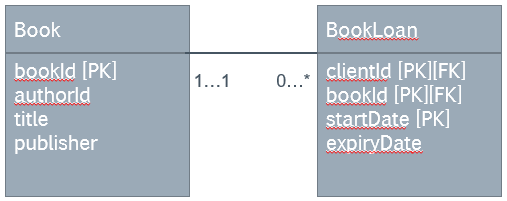
**Question 2**

We are still dealing with an online book store where you can both buy and borrow books, electronic books and audiobooks. Here, we study the relationship between the BookLoan table and the Book relation.



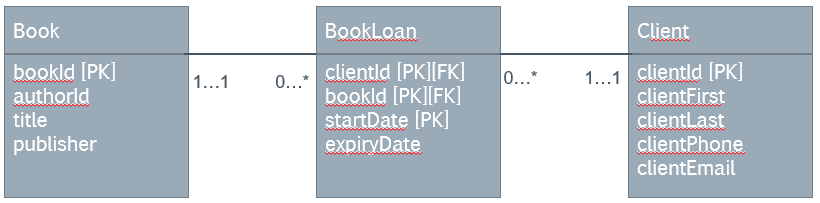
Which of the following options represents the relationship most accurately?

A screenshot of a computer

Description automatically generated

**Question 3**

Having developed the relationships between the related tables, we have the following diagram:

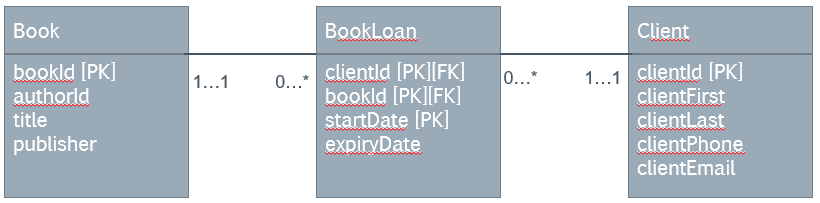


What, if anything, is wrong with this design?

* Nothing. BookLoan is the weak entity that models an activity in which the strong entities Book and Client are involved.

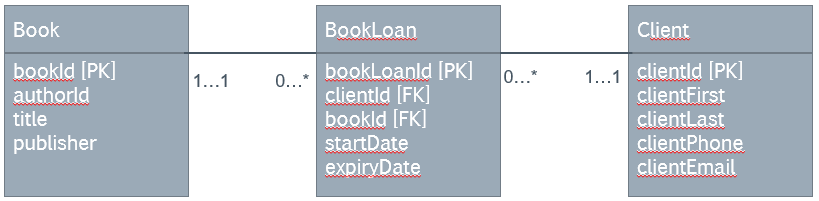
**Question 4**

The tables and relationships that enable us to take out books for a loan look like this (the same as in Q 3):



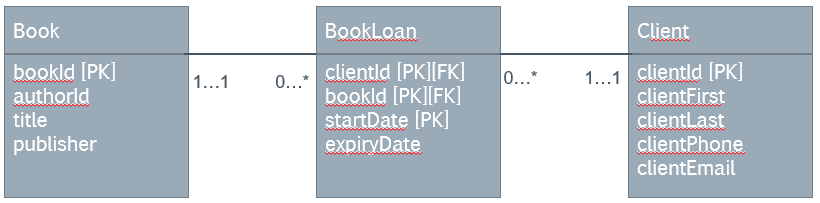
Some of us think that a composite key is cumbersome, and want to replace it with a surrogate key. What are these tables going to look like if we replace the composite key with a surrogate key?

A screenshot of a computer

Description automatically generated

**Question 5**

The tables and relationships that enable us to take out books for a loan look like this (the same as in Q 3):

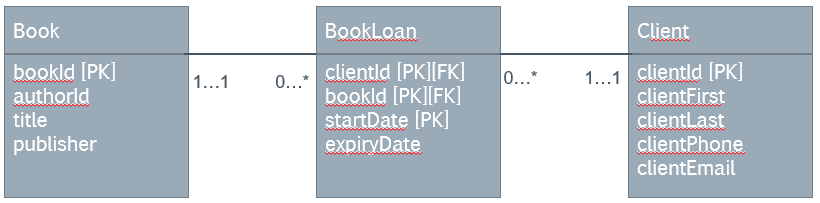


Until now, we did not allow extensions on loans. We have a new requirement that our database has to record extensions (loan renewals). Does the current design allow us to record extensions without losing the history of the loan and due date?

* Yes, but with difficulty. We can make a new loan entry for the same book and client with new dates. To find out whether it is a new loan or the continuation of a current one, we have to compare start and expiry dates of subsequent entries.

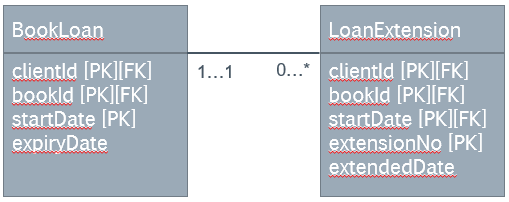
**Question 6**

The tables and relationships that enable us to take out books for a loan look like this (the same as in Q 3):



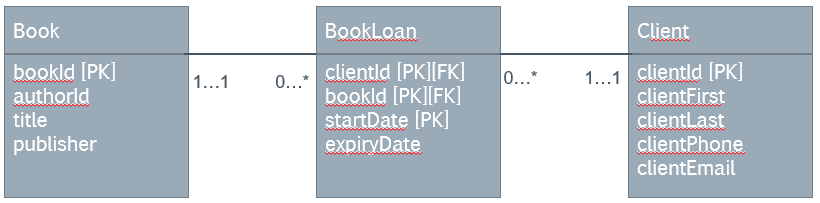
We are still debating what would be the best design to accommodate loan renewals (extensions). We have received new information to the effect that we have to ensure the extensions can be traced to the original loan, because the store wants to enforce a limit on the number of extensions. Which of the following is the best design for such a requirement? (The strong entities Book and Client have been omitted in the answers.)

A screenshot of a computer

Description automatically generated

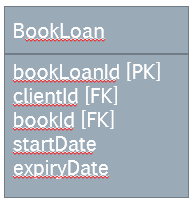
**Question 7**

The tables and relationships that enable us to take out books for a loan look like this (the same as in Q 3):



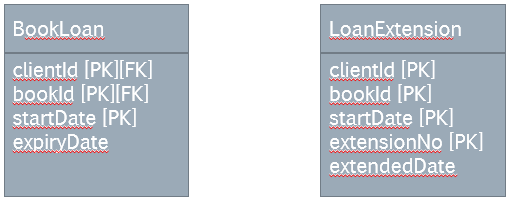
We are interested in the BookLoan table. Now that we have to create a child table to enable extensions of loans, we are considering using a surrogate key for BookLoan. What would the BookLoan table possibly look like if we used a surrogate key instead of the natural composite key?

A screenshot of a computer

Description automatically generated

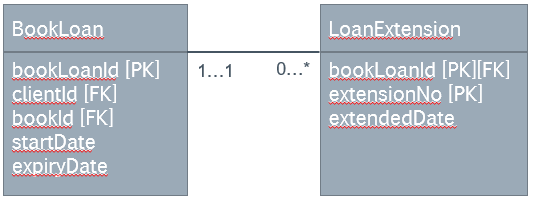
**Question 8**

Our client has decided that the limit on loan extensions is a very useful feature. Based on their requirements, we have designed the following relations (other relations exist but have been omitted, but the foreign keys to the Book and Client tables are included here).



We have not completed the design yet, the relationship between the two tables is still missing. We are contemplating the removal of the composite key from BookLoan and its replacement with a surrogate key. What would both relations and their relationship look like, if we used a surrogate key for the BookLoan only (retaining the composite key of LoanExtension)? As per the relational design principles, we do not retain any redundant attributes.

A screenshot of a computer

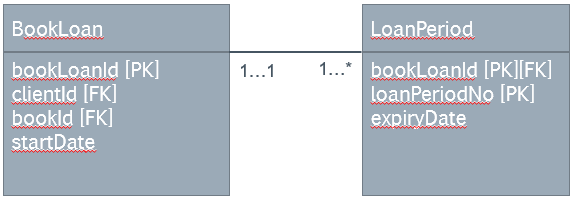
Description automatically generated

**Question 9**

Unlike in the earlier scenario, the book store owner now says that there is no difference between the first loan period and the extensions. They are all just loan periods. The store owner just wants the loan periods stored for each loan, because there is a limit of loan periods after which the client has to buy the book.

What could our BookLoan and LoanPeriod tables look if we are to satisfy this requirement?

A screenshot of a computer

Description automatically generated

**Question 10**

The same book is often published several times, and potentially by different publishers. One publishing company naturally publishes a great number of books. How can we model this relationship?

* This is a many-to-many relationship. We have to resolve it using a weak entity, for example:

