Consider the following relations of a database for a Library:

Book (ISBN, PublicationDate, Pages Integer, Title, AuthorID)
Author (AuthorID , AuthorFirstName, AuthorLastName)
Categories (CategoryID, CategoryName)
BookCategories (ISBN, CategoryID)
Members (MemberID, MemberName, MemberPhone, MemberDate, MemberEmail)
Loans (MemberID, ISBN, LoanDate, DueDate, ReturnDate, LateReturnFine)

Where:

Primary keys are underlined,

Book.AuthorID is a foreign key to Author.AuthorID,

BookCategories.ISBN is a foreign key to Book.ISBN,

BookCategories.CategoryID is a foreign key to Categories.CategoryID,

Loans.ISBN is a foreign key to Book.ISBN,

Loans.MemberID is a foreign key to Members.MemberID.

Write SQL update statements to perform the following operations:

1. Add the book "Pan" by Knut Hamsun, published on 2014-02-16. The book has 134 pages and ISBN 9781495968099. (5 pts)

2. Update all the late return fines that are less than $5 and set them to $0. (5 pts)

3. Delete all the books that have never been loaned by any member of the library so far. (9 pts)

Write SQL statements to create the following views:

4. A view that contains the first and last name of authors, name of book categories and number of books that each author published in that category. (10 pts)

5. A view that contains the ISBN and title of all the books from the category "Novel", and the number of times each book was loaned. (10 pts)

6. A view that contains the names of all the members of the library, number of books they have loaned and total amount of fine due to late return. (10 pts)

7. Considering the following view, state which of the following queries and updates would be

allowed on the view. If a query or update would be allowed, give its result when applied to the database; If not allowed, give the reason.

```
CREATE VIEW CategoryCount
   AS SELECT MemberName, CategoryName,
   COUNT(*) as LoanCount
   FROM Members, Categories, BookCategories, Loans
   WHERE Loans.MemberID = Members.MemberID
      AND Loans.ISBN = BookCategories.ISBN
      AND BookCategories.CategoryID = Categories.CategoryID
   GROUP BY MemberName, CategoryName
```

a) SELECT DISTINCT MemberName FROM CategoryCount
   WHERE LoanCount = 4 (4 pts)

b) SELECT * FROM CategoryCount
   WHERE CategoryName = "Fantasy"
   AND LoanCount > 2 (4 pts)

c) UPDATE CategoryCount
   SET LoanCount =0
   WHERE CategoryName = "Action" (4 pts)

Write SQL statements to answer the following queries:

8. Find names of the members who have read all the books written by "Gabriel Marquez" (AuthorID = 14). (13 pts)

9. Find the names of all the members who have loaned the "Fantasy" book with maximum number of pages. Show also the loan date and return date. (13 pts)

10. Find the name of the authors and titles of their books which belong to exactly one category. (13 pts)