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

```
INSERT INTO Books
VALUES ('9781495968099', '2014-02-16', 134, 'Pan', 8);
```

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

```
UPDATE Loans
SET LateReturnFine=0
WHERE LateReturnFine < 5;
```

For each book, the subquery selects all loans records of that book. If the result set is empty, it means that the book have never been loaned by any member and is therefore selected.

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

```
DELETE FROM Books
        WHERE NOT EXISTS (SELECT * FROM Loans
                                WHERE Books.ISBN = Loans.ISBN);
```

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.

```
CREATE VIEW AUTHORING_INFO
AS SELECT AuthorFirstName, AuthorLastName, CategoryName, COUNT(*) AS
Total_Published
FROM ((Books NATURAL JOIN Author) NATURAL JOIN BookCategories)
        NATURAL JOIN Categories
GROUP BY AuthorID, CategoryID;
```

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.

```
CREATE VIEW NOVEL_INFO
AS SELECT ISBN Book_ISBN, Title Book_title, COUNT(LoanDate) Total_loaned
FROM ((Books LEFT NATURAL JOIN Loans) NATURAL JOIN BookCategories)
NATURAL JOIN Categories
WHERE CategoryName='Novel'
GROUP BY ISBN;
```

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.

```
CREATE VIEW Loans_Statistics
AS SELECT MemberName, COUNT(LoanDate) Total_loaned, SUM(LateReturnFine)
Total_fine
FROM Members LEFT NATURAL JOIN Loans
GROUP BY MemberID
```

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.

a) SELECT DISTINCT MemberName FROM CategoryCount WHERE LoanCount=4

MemberName
Betty Ford
Debra Bryant
Kenneth Armstrong

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

MemberName	CategoryName	LoanCount
Debra Bryant	Fantasy	4
Ralph Schmidt	Fantasy	3
Raymond Carter	Fantasy	3

c) UPDATE CategoryCount SET LoanCount =0
WHERE CategoryName = “Action"

This update is not allowed, since CategoryCount is a view that uses grouping and aggregate functions such as “GROUP BY” and “COUNT”. Views defined using grouping and aggregate functions are not updatable. Also, views defined on multiple tables using joins which generally not updatable.

First subquery selects all books written by “Gabriel Marquez” (AuthorID=14), and the second subquery (which is correlated) selects all books that the particular member have read. If the set difference of the first subquery result MINUS (EXCEPT) the second subquery result is empty, it means that the member reads all the books written by Gabriel Marquez and is therefore selected.

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

```
SELECT MemberName
FROM Members M
WHERE NOT EXISTS (SELECT ISBN FROM Books WHERE AuthorID=14
                  EXCEPT
                  SELECT DISTINCT ISBN
                  FROM Loans
                  WHERE M.MemberID=MemberID);
```

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.

```
SELECT MemberName, LoanDate, ReturnDate
FROM Members NATURAL JOIN Loans
WHERE ISBN IN (SELECT ISBN
        FROM (Books NATURAL JOIN BookCategories) NATURAL JOIN Categories
        WHERE CategoryName='Fantasy' AND
         Pages = (SELECT MAX(Pages)
          FROM (Books NATURAL JOIN BookCategories) NATURAL JOIN Categories
          WHERE CategoryName='Fantasy' ))
```

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

```
SELECT AuthorFirstName, AuthorLastName, Title
FROM Author NATURAL JOIN Books
WHERE ISBN IN (SELECT ISBN
                  FROM BookCategories
                  GROUP BY ISBN
                  HAVING COUNT(DISTINCT CategoryID) = 1);
```