**Design a normalized database schema for the following requirement.**

The requirement: A library wants to maintain the record of books, members, book issue, book return, and fines collected for late returns, in a database. The database can be loaded with book information. Students can register with the library to be a member. Books can be issued to students with a valid library membership. A student can keep an issued book with him/her for a maximum period of two weeks from the date of issue, beyond which a fine will be charged. Fine is calculated based on the delay in days of return. For 0-7 days: Rs 10, For 7 – 30 days: Rs 100, and for days above 30 days: Rs 10 will be charged per day.

**Sample Database Design**

STUDENT(Student\_id, Name, Deptid, Email\_Id, Phone\_Number) Deptid is FK

Department(Department\_id, Name)

BOOK (Book\_Id, Title, Language\_Id, MRP, Publisher\_Id, Published\_Date, Volume, Status) //

Language\_Id, Publisher\_Id are FK (Foreign Key)

AUTHOR(Author\_Id, Name, Email, Phone\_Number, Status)

BOOK\_AUTHOR(Book\_Id, Author\_Id) // many-to-many relationship, both columns are PKFK

(Primary Key and Foreign Key)

PUBLISHER(Publisher\_id, Name, Address)

MEMBER(Member\_Id, Name, Roll\_Number, Date\_of\_Join,Date\_of\_Expiry, Status)

BOOK\_ISSUE(Issue\_Id, Date\_Of\_Issue, Book\_Id, Member\_Id, Expected\_Date\_Of\_Return,

Status) // Book\_Id and Member\_Id are FKs

BOOK\_RETURN(Issue\_Id, Actual\_Date\_Of\_Return, LateDays, LateFee) // Issue\_Id is PK and

FK

LANGUAGE(Language\_id, Name) //Static Table for storing permanent data

LATE\_FEE\_RULE(FromDays, ToDays, Amount) // Composite Key

***EXERCISES***

*1. Create a normalized database design with proper tables, columns, column types, and*

*constraints*

*2. Create an*[*ER diagram*](http://examonline.today/mod/assign/view.php?id=783)*for the above database design.*

*3. Write SQL commands to*

a. Create DDL statements and create the tables and constraints (from the design) in the

database created in step-a (Library)

Notes: [ Create a script file and execute it. Create the script file in such a way that, if the

table exists, drop the tables and recreate )]

b. Create and execute DROP TABLE command in tables with and without FOREIGN

KEY constraints.

c. Create and execute ALTER TABLE command in tables with data and without data.

d. Create and execute SQL commands to build indices on Member\_Id and Book\_Id on

table Book\_Issue.

e. Create and execute GRANT/REVOKE commands on tables.

f. Create and execute SQL commands to insert data into each of the tables designed

g. Create and execute UPDATE/DELETE commands on tables. Try to update/delete

rows with Primary and Foreign Keys. Try bulk updates or deletes using SQL

UPDATE statement

*4. Write SQL Query to retrieve the following information*

a. Get the number of books written by a given author

b. Get the list of publishers and the number of books published by each publisher

c. Get the names of authors who jointly wrote more than one book.

d. Get the list of books that are issued but not returned

e. Get the list of students who reads only ‘Malayalam’ books

f. Get the total fine collected for the current month and current quarter

g. Get the list of students who have overdue (not returned the books even on due date)

h. Calculate the fine (as of today) to be collected from each overdue book.

i. Members who joined after Jan 1 2021 but has not taken any books

*5. Book return should insert an entry into the Book\_Return table and also update the status in Book\_Issue table as ‘Returned’. Create a database TRANSACTION to do this operation (stored procedure).*

*6. Create a database view ‘Available\_Books’, which will list out books that are currently*

*available in the library*

*7. Create a database procedure to add, update and delete a book to the Library database (use parameters).*

*8. Use cursors and create a procedure to print Books Issue Register (page wise – 20 rows in a page)*

*9. Create a history table (you may use the same structure without any keys) for the MEMBER table and copy the original values of the row being updated to the history table using a TRIGGER.*

**Basic Questions:**

Problem # 1:

Write a query to display the member id, member name and membership status who are all having

life time membership. Hint: Life time membership status is  “Active”.

Problem # 2:

Write a query to display the member id, member name who have not returned the books. Hint: Book

return status is book\_issue\_status ='issed' or 'not issued' or ‘returned’.

Problem # 3:

Write a query to display the member id, member name who have taken the book with book code 'B1101'.

Problem # 4:

Write a query to display the book code, book title and author of the books whose author name begins with 'H'

Problem # 5:

Write a query to display the total number of  books  by  each author available in library with alias name ‘NO\_OF\_BOOKS’.

Problem # 6:

Write a query to display the number of books published by "dcbooks” with the alias name

“NO\_OF\_BOOKS”.

Problem # 7:

Write a query to display the book code, book title of the books which are issued on the date "08-Mar-2021".

Problem # 8:

Write a query to display the member id, member name, date of registration and expiry date of the members whose membership expiry date is before APR 2021.

Problem # 9:

write a query to display the member id, member name, date of registration, membership status of the members who registered before "March 2021" and membership status is "Temporary"

Problem # 10:

Write a query to concatenate book title, author and display in the following format.

Book\_Title\_is\_written\_by\_Author

Example: Let Us C\_is\_written\_by\_Yashavant Kanetkar

Hint: display unique books. Use “BOOK\_WRITTEN\_BY” as alias name.