Activity 2. Translate the following model of an online library into tables expressed in relational graphs and create them, using the DDL SQL language of Oracle. Use the data type that you consider most appropriate.

Books (ISBN, title, authorName, genre)

Primary Key: ISBN

Non-Null Value: Title

Copies (CopieCode, BookISBN, edition)

Primary Key: CopieCode

Foreign Key: BookISBN -> Books

Members (MemberCode, DNI, name, registrationDate, country, email)

Primary Key: DNI

Alternate Key: MemberCode

Loans (CopieCode, MemberCode, loanDate, returnDate)

Primary Key: CopieCode, MemberCode

Foreign Key: CopieCode -> Copies

Foreign Key: MemberCode -> Members

The membership code will be associated with a sequential name seq\_members, which will also have to be created.

Activity 3. On the previous tables, perform the following operations:

a) Modify the “members” table so that the primary key is cod\_socio, instead of dni.

b) Modify the “copies” table by adding the year of publication as an attribute, which must not be less than 2000 or greater than 2023.

c) Modify the “members” table by adding an attribute that allows you to store the member's address and another that stores the scanned image of the book cover.

Activity 4. On the tables created in the previous activity, perform the following operations using transactions:

a) Insert 5 records into the books table, one of them with the following isbn 978-1-56619-909-4

b) Insert 3 records into the copies table, one of them with code 101.

c) Insert 5 records into the members table, one of them with ID number 98765432B.

d) Insert 5 records into the loans table, one of them with the copy with code 101

e) Update the title of the book with isbn 978-1-56619-909-4 to “One hundred years of solitude”.

f) Update the name of the member with DNI 98765432B to María López.

g) Update the loan return date of the copy with code 101 to 01-06-2023.

Activity 5. On the previous DB, perform the following queries:

a) Show the information of all the copies in the library, and for those that have ever been borrowed, the code of the member that has borrowed the copy.

b) Obtain the average loan duration (in days) of the most borrowed book of all in the summer of 2024, encapsulate it in a view called top\_books.

c) Obtain the member who has made the most loans during the week from the members with a reported country and a Gmail email. Sort the results by member code. If the member's name is not reported, return 'No\_name'.

d) Obtain the list of members who have never made a loan using the clause EXISTS or a variant thereof.

e) Obtain the name of the members who have borrowed books from the ALFAGUARA edition - that have not been borrowed more than 5 times- during the summer months of the year 2022 and have made at least one other loan in the same year.

f) Insert a new loan into the loan table. The CodExemplar will be the one with the lowest number of loans so far, of the copies that belong to books of the comedy genre (if there is more than one we will keep the first one in alphabetical order by title), the CodMember will be the member who has made the most loans of all in the year 2023 (if there is more than one we will keep the first one in alphabetical order by name). FechaPrestamo will be reported with the system date and fechaDevolucion will not be filled.

Activity 6. On the previous BBDD, perform the following administration operations:

a) Create a user dam\_usuari with query permissions on all tables of the database.

b) Create a user dam\_gestor with insert and update permissions on the table loans.

c) Revoke the permissions of the user dam\_usuari on the table books.

d) Delete both users.

e) Create an index for the nomAutor field of the table books.

Activity 7. Create a stored procedure in PL/SQL that allows to register a new loan in the loan table. The procedure must receive as parameters the item code, the member code and the loan date. Before inserting the new loan, the procedure must verify whether the item is available (that is, if there is no loan that does not have a return date reported). If the item is available, the new loan must be inserted into the loan table. If it is not available, the procedure must display a message indicating that the item is not available for loan. Perform exception handling and transactional control. Provide an anonymous code block to test the subroutine.

Activity 8. Create a PL/SQL function that receives a member code as a parameter and returns the duration of the longest loan made by that member in days. The loan duration will be calculated by subtracting the return date from the loan date. If the member has no loans registered, the function must return 0. Provide an anonymous code block to test the subroutine.

Activity 9. Create a PL/SQL trigger that fires when a new record is inserted into the Loans table. This trigger should automatically update the loan due date by adding 7 days to the loan date. If the due date is already specified in the insert, the trigger should not perform any action. Provide the DML operation that will fire the trigger.