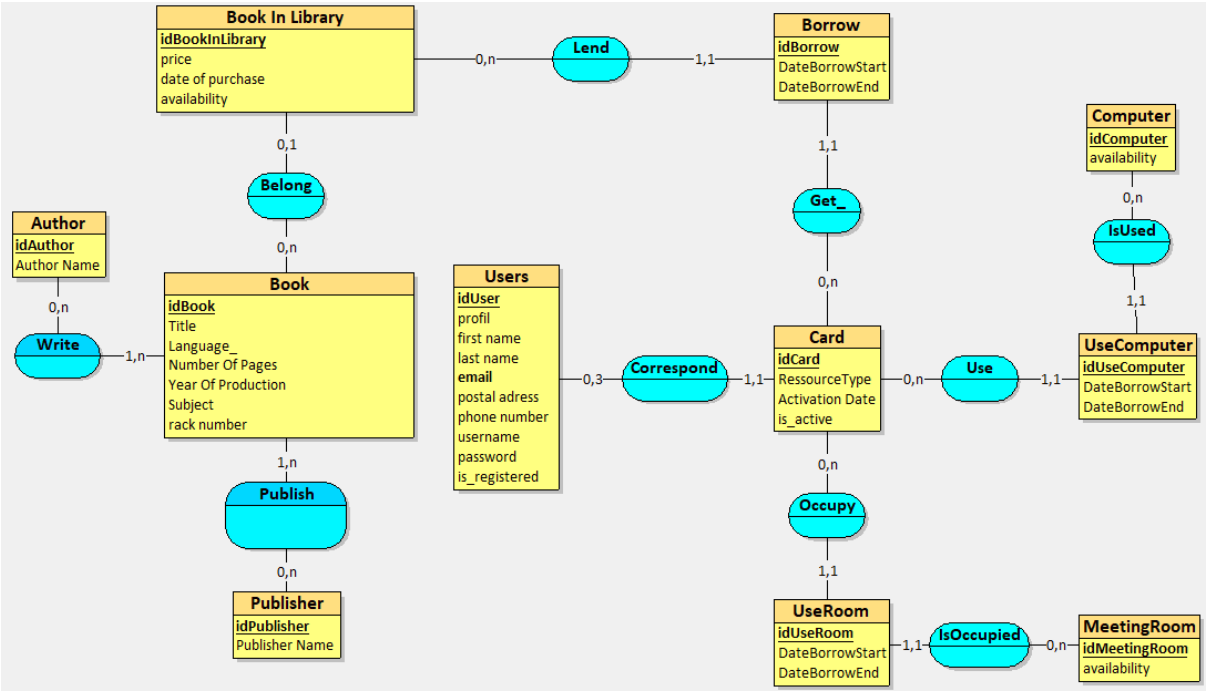
Project: Construction and exploitation of a database to automate the services of the library of a university

Creation of the database

Conceptual Data Model (CMD)



To create our Conceptual Data Model, we have used the software looping.

We have made a few suppositions to design it:

-We suppose that a book can only have one subject (it is its main subject).

-Each user can have at most 3 cards and each card must have a different type, this is why the cardinality of the relation between Users and Card is (0,3).

-Our entity Card has all types of cards whose types are determined by the attribute RessourceType. We distinguish which card should be used for what by using triggers.

Logical Model of the Database(LMD)

Book = (**idBook**, Title, Language, Number_Of_Pages, Year_Of_Production, Subject, rack_number)

Write = (**#idBook, #idAuthor**)

Author = (**idAuthor**, Author_Name)

Publish = (**#idBook, #idPublisher**)

Publisher = (**idPublisher**, Publisher_Name)

Book_In_Library = (**idBookInLibrary**, price, date_of_purchase, availability, **#idBook**);

Borrow = (**idBorrow** , DateBorrowStart, DateBorrowEnd , #idCard, #idBookInLibrary);

Card = (**idCard**, RessourceType , Activation_Date, is_active , #idUser);

Users = (**idUser**, profil , first_name , last_name , email , postal_adress , phone_number , username , password , is_registered)

Computer = (**idComputer**, availability);

UseRoom = (**idUseRoom** , DateBorrowStart, DateBorrowEnd, #idCard, #idMeetingRoom);

MeetingRoom = (**idMeetingRoom** , availability);

UseComputer = (**idUseComputer** , DateBorrowStart, DateBorrowEnd, #idCard, #idComputer);

Computer = (**idComputer**, available);

Structural Constraints of the database:

NAME	TYPE	CONSTRAINCTS	ENTITIES	DESCRIPTION
idBook	*INT*	*NOT NULL, UNIQUE, PRIMARY KEY*	*Book, AUTO INCREMENT*	
Title	VARCHAR	NOT NULL	Book	
Language	VARCHAR	NOT NULL	Book	
Number_Of_Pages	INT	NOT NULL	Book	
Year_Of_Production	DATE	NOT NULL	Book	
Subject	VARCHAR	NOT NULL	Book	
rack_number	INT	NOT NULL	Book	
idAuthor	INT	*NOT NULL, UNIQUE, PRIMARY KEY*	*Author, AUTO INCREMENT*	
Author_Name	VARCHAR	NOT NULL	Author	
idPublisher	INT	*NOT NULL, UNIQUE, PRIMARY KEY*	*Publisher, AUTO INCREMENT*	
Publisher_Name	VARCHAR	NOT NULL	Publisher	
idStudent	INT	*NOT NULL, UNIQUE, PRIMARY KEY*	*Users, AUTO INCREMENT*	
profil	VARCHAR	NOT NULL CHECK IN (‘Administrator’,'Library Agent', 'Student')		
first_name	VARCHAR	NOT NULL	Users	
last_name	VARCHAR	NOT NULL	Users	
email	VARCHAR	NOT NULL, UNIQUE	Users	
postal_adress	VARCHAR	NOT NULL	Users	
phone_number	VARCHAR	NOT NULL	Users	
is_registered	BOOLEAN	NOT NULL	Users	
username	VARCHAR	NOT NULL, UNIQUE	Users	
password	VARCHAR	NOT NULL	Users	
idBookInLibrary	INT	*NOT NULL, UNIQUE, PRIMARY KEY*	*BookInLibrary, AUTO INCREMENT*	
price	INT	NOT NULL	BookInLibrary	
date_of_purchase	DATE	NOT NULL	BookInLibrary	
idBook	INT	NOT NULL FOREIGN KEY	BookInLibrary	
idCard	INT	*NOT NULL, UNIQUE, PRIMARY KEY*	*Card, AUTO INCREMENT*	
RessourceType	VARCHAR	NOT NULL CHECK IN (‘Book’,’Computer’,’Meeting_Room’)	Card	
Activation_Date	DATE	NOT NULL	Card	
is_active	BOOLEAN	NOT NULL	Card	
idUser	INT	NOT NULL FOREIGN KEY	Card	
idBook	INT	*NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY*	*Write_*	
idAuthor	INT	*NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY*	*Write_*	
idBook	INT	*NOT NULL, UNIQUE,*	*Publish*	

		PRIMARY KEY, *FOREIGN KEY*		
idPublisher	INT	*NOT NULL, UNIQUE,* *PRIMARY KEY,* *FOREIGN KEY*	*Publish*	
idBorrow	INT	*NOT NULL, UNIQUE,* *PRIMARY KEY,*	*Borrow*	
idBookInLibrary	INT	NOT NULL, FOREIGN KEY	Borrow	
idCard	INT	NOT NULL, FOREIGN KEY	Borrow	
DateBorrowStart	DATE	NOT NULL	Borrow	
DateBorrowEnd	DATE	NOT NULL	Borrow	
idUserRoom	*INT*	*NOT NULL, UNIQUE,* *PRIMARY KEY*	*UserRoom*	
DateBorrowStart	DATE	NOT NULL	UseRoom	
DateBorrowEnd	DATE	NOT NULL	UseRoom	
idCard	INT	NOT NULL, FOREIGN KEY	UseRoom	
idMeetingRoom	INT	NOT NULL	UseRoom	
idUserComputer	*INT*	*NOT NULL, UNIQUE,* *PRIMARY KEY*	*UseComputer*	
DateBorrowStart	DATE	NOT NULL	UseComputer	
DateBorrowEnd	DATE	NOT NULL	UseComputer	
idCard	INT	NOT NULL, FOREIGN KEY	UseComputer	
idComputer	INT	NOT NULL	UseComputer	
idComputer	*INT*	*NOT NULL, UNIQUE,* *PRIMARY KEY*	*MeetingRoom*	
availability	BOOLEAN	NOT NULL	MeetingRoom	
idComputer	*INT*	*NOT NULL, UNIQUE,* *PRIMARY KEY*	*Computer*	
availability	BOOLEAN	NOT NULL	Computer	

List of triggers:

-trigger to delete the Books of a deleted Author
-trigger to delete the Books of a deleted Publisher
-trigger to check it a user can have a new card or not

-trigger to check if a card can be used to borrow books
-trigger to check if a card can be used to borrow a computer
-trigger to check if a card can be used to borrow a room

-trigger to check if a user can borrow a book or not, a user can borrow a book if the book is available and the user has borrowed less than 5(or 1 if not registered to the university) books.
-trigger to check if the user can borrow a room or not (check only the availability of the room)
-trigger to check if the user can borrow a computer or not (check only the availability of the computer)

-trigger to change the availability of a book after being borrowed
-trigger to change the availability of a book after being returned
-trigger to check for books that DateBorrowStart isn't superior to DateBorrowEnd

-trigger to change the availability of a room after being borrowed
-trigger to change the availability of a room after being returned
-trigger to check for rooms that DateBorrowStart isn't superior to DateBorrowEnd

-trigger to change the availability of a computer after being borrowed
-trigger to change the availability of a computer after being returned
-trigger to check for computers that DateBorrowStart isn't superior to DateBorrowEnd

List of users:

root:

- Select, Insert, and Update on all tables

student:

- Select on Users, Card, Borrow, UseRoom, UseComputer, BookInLibrary, Book, Computer, MeetingRoom tables
- Update on Users

library_agent:

- Select on all tables
- Insert and Update on tables Borrow, UseRoom, UseComputer

login:

- Select and Insert on table Users