DATABASE LIBRARY MANAGEMENT SYSTEM



GROUP 4

ĐẶNG VIỆT ANH | HE151359

TRẦN XUÂN QUANG | HE153654

ĐOÀN HẢI PHONG | HE151364

1. Introduction Problem
2. Describe Problem

Nowaday, libraries were faced with an information explosion and the rapid growth rate of their collections. The issues confronting library administrators during that period were mainly physical managements involving shelving and weeding of materials, storage space, users’ in-house access to the collection, and preservation of the materials. After our team researched, the results are as follows:

- Each person who borrows books needs to have a separate account to manage information instead of checking the physical library account, that information includes necessary personal information such as phone number, address, gender, etc. count, date of birth...

-Each employee of the library also has a separate account to manage the borrowing of books from the library, and has more information of the same staff as the person who borrows the book.

-Books are categorized by publisher and book genre, one producer can produce many books and one genre can have many books

-The book loan voucher will have a link between an employee and a borrower to represent each time a book is borrowed containing information about the date of borrowing.

-The details of the loan slip will store information about the loan slip, borrowed books, number of books borrowed and return date

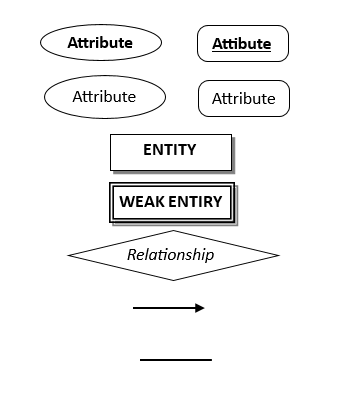
1. Goal

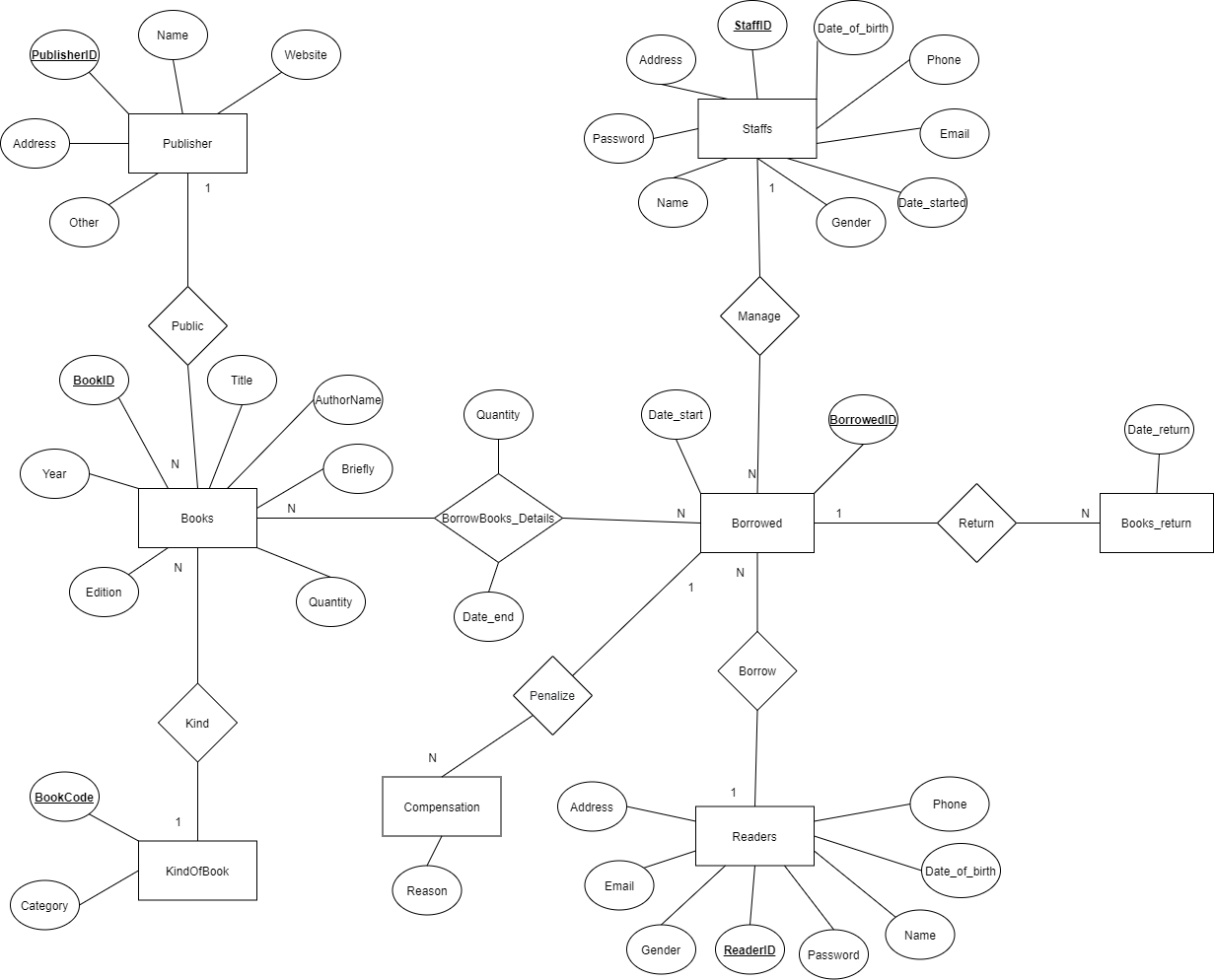
This database is written to solve the stages and procedures of the library by retrieving information on the computer without the need for manual work as before. This helps both library managers and readers easily control the information of books as well as the return of books in the library.

1. ER Diagrams and data scheme

1. SET-UP ENTITY – RELATIONSHIP

Some symbols used in the model :

* Key / identifier attribute
* Attribute description / description
* Entity
* Weak entity
* Relationship
* Connectivity (force) = 1
* Connectivity = N
* 



1. Data Scheme

**Publisher**(PublisherID, Name, Address, Website, Other)

**KindOfBook**(BookCode, Category)

**Books**(BookID, BookCode, PublisherID, Title, AuthorName, Year, Edition, Quantity, Brief)

**Staffs**(StaffID, Password, Name, Date\_of\_birth, Address, Gender, Phone, Email, Date-started)

**Readers**(ReaderID, Password, Name, Date\_of\_birth, Address, Gender, Email)

**Borrowed**(BorrowedID, StaffID, ReaderID, Date\_start)

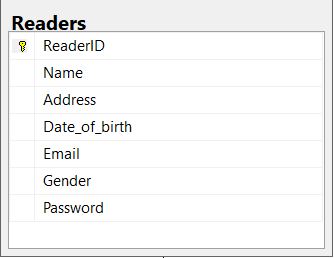
**BorrowBooks\_details**(BorrowID,BookID, Quantity, Date\_end)

**Books\_return**(BorrowID,Date\_return)

**Compensation**(BorrowedID, Reason)



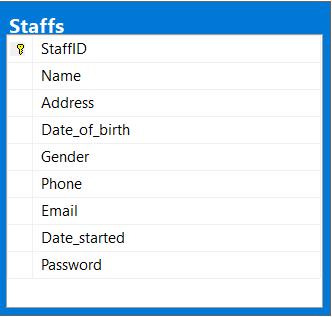
A.Readers



This is the entity Readers,this has 7 attributes.

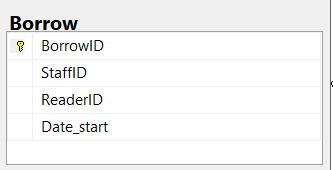
Each readers have ReaderID as ID ( primary key) .Context of reader has Email,Address.Reader’s information has Name,date\_of\_birth,gender,password.

B.Staffs



This is the entity Staffs,this has 9 attributes.Each staff have StaffID as identity(primary key).Context of staffs have address ,phone,email.Staff’s information has Name,date\_of\_birth,gender,date\_started,password.Context of staffs has address,phone,email.

C.Borrow



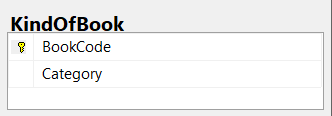
This is the entity Borrow,this has 4 attributes.Each record of borrow have BorrowID(primary key),StaffID(Foreign key from table Staffs),ReaderID(Foreign key from table Readers),Date\_start(the day start borrow)

D.Compensation



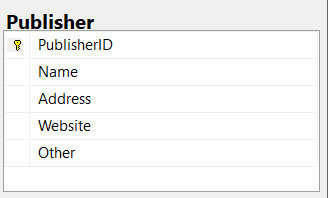
This is the entity Compensation,this has 2 attributes.Each compensation have BorrowID(foreign key references from table Borrow),reason is the reason to borrow book(purpose).

E.KindOfBook



This is the entity KindOfBook,this has 2 attributes.Each kind of book have BookCode as ID of kind (primary key),Category(name of kind).

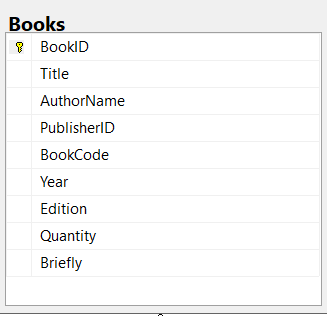
G.Publisher



This is the entity Publisher,this has 5 attributes.Each Publisher have PublisherID as ID of publisher (foreign key),Name(name of publisher),Address(address of publisher),

Website(context online),Other(others information).

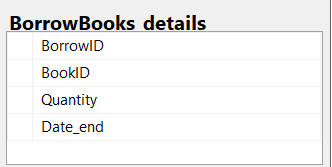
H.Books



This is the entity Books(information of all books in library),this has 9 attributes.Each book have BookID as ID of book (primary key).Book’s information have Title(Name of books),AuthorName(name of author).Information about the publishing process have PublisherID(foreign key from Publisher),BookCode(foreign key from KindOfBook)

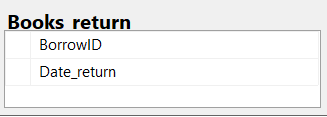
year(the year publishing),Edition,Quantity(The amount publishing),Briefly(key content).

I.BorrowBooks details



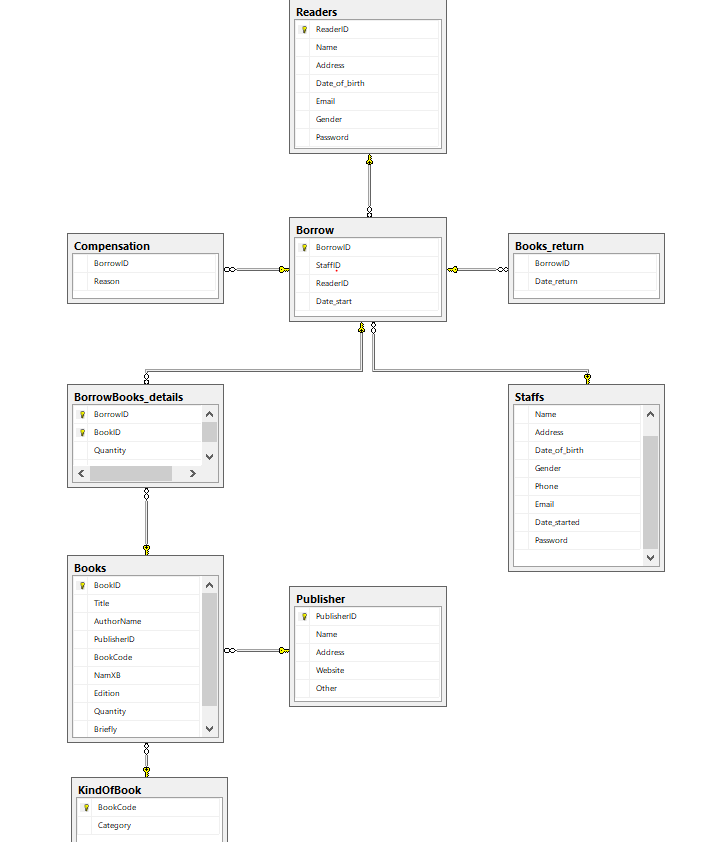
This is the entity BorrowBooks details(record borrowing book),this has 4 attributes.Each record have BorrowID as ID of record(foreign key from Borrow), BookID as ID of book(foreign key from Books),Quantity(the amount of book in 1 time borrow),Date\_end(the day return book)

J.Books return



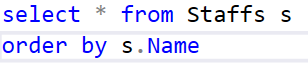
This is the entity Books return(information record about returning book),this has 2 attributes .Each record have BorrowID as ID of record(foreign key from Borrow),Date\_return(the day return book of readers)

Full Diagram

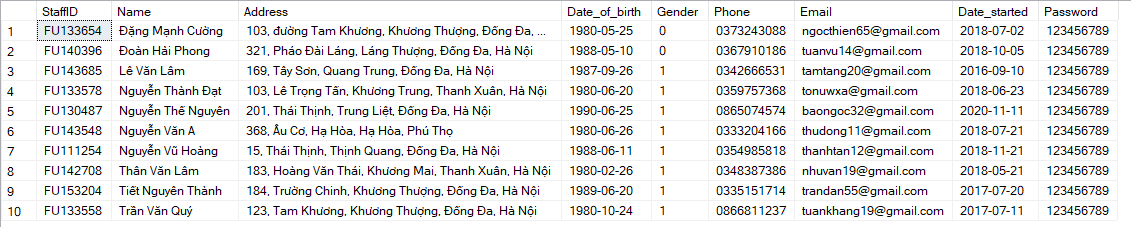


1. SQL command
2. Query using order by

Code :



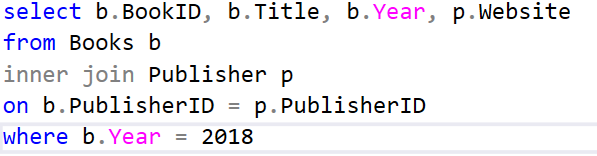
Test case :



We use **ORDER BY** to sort the list ascending by informations of staffs

1. Query using join

Code :



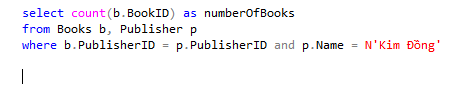
Test case :



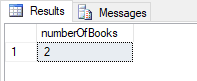
We use **INNER JOIN** to select book publish in 2018 and its website

1. Query using aggregate functions

Code :

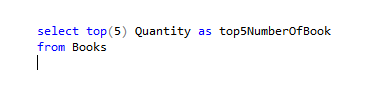


Test case :

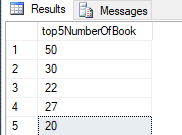


We use function **COUNT()** with parameter BookID to count the number of book has publisher name is Kim Đồng

Code :



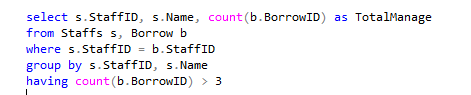
Test case :



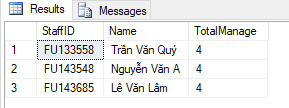
We use function **TOP()** to select top 5 with the most number of books

1. Query using group by and clauses

Code :



Test case :



We use **GROUP** BY and **HAVING** clauses to count staffs manage more than 3 bills

1. Query using subquery as a relation

select b.BookID, b.Title, count(br.ReaderID) as TotalBorrows

from Books b, BorrowBooks\_details bd, Borrow br, Readers r

where b.BookID = bd.BookID and bd.BorrowID = br.BorrowID and br.ReaderID = r.ReaderID

group by  b.BookID, b.Title

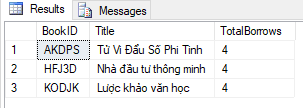
having count(br.BorrowID) = (select top(1) count(br.ReaderID) as TotalBorrows

from Books b, BorrowBooks\_details bd, Borrow br, Readers r

where b.BookID = bd.BookID and bd.BorrowID = br.BorrowID and r.ReaderID = br.ReaderID

group by  b.BookID, b.Title

order by count(br.BorrowID) desc)



We using subquery to find the most book borrowed by reader and get in into having clause

select r.ReaderID, r.Name, sum(bd.Quantity) as NumberOfBorrowedBooks

from Readers r, Borrow b, BorrowBooks\_details bd

where r.ReaderID = b.ReaderID and b.BorrowID = bd.BorrowID

group by r.ReaderID, r.Name

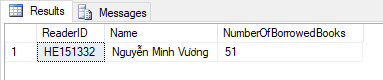
having sum(bd.Quantity) = (select top(1) sum(bd.Quantity) as NumberOfBorrowedBooks

from Readers r, Borrow b, BorrowBooks\_details bd

where r.ReaderID = b.ReaderID and b.BorrowID = bd.BorrowID

group by r.ReaderID, r.Name, bd.BorrowID

order by sum(bd.Quantity) desc)



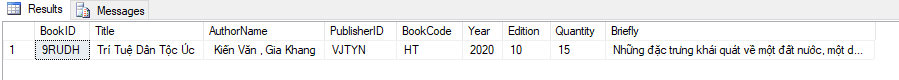
We use subquery to find the reader borrowed most book and get it into having clause

F. QUERY THAT USES PARTIAL MATCHING IN THE WHERE CLAUSE

select \*

from Books

where Title like N'%Trí Tuệ%' and Quantity > 10



We use PARTIAL MATCHING in the WHERE clauses to combine binding conditions

**g.**     **query that uses a self – join**

***Code:***

SELECT s.StudentID, s.LastName +' '+ s.[First Name] AS 'Full Name', r.[Check Price]

FROM dbo.STUDENT s, dbo.REGISTER r

WHERE r.StudentID = s.StudentID AND r.[Check Price] = 1

**A.**     **store procedure**

create proc check\_Quantity @Book\_ID char(10), @NumberofBooks int output

as

begin

set @NumberofBooks = (select Quantity

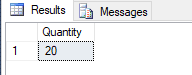
from Books where BookID = @Book\_ID)

end

declare @t int

exec check\_Quantity'054JD', @t output

select @t as Quantity



We use procedure to check the quantity of books for which the book's ID is entered

by the librarian or user

**A.**     **trigger**