**Reading Book Management**

I. Use case.

Users

Manager

* For Users:

+ Book : users can read book

+ Rate: rate for books that user read

+ Active : view activate

+ Information: update information

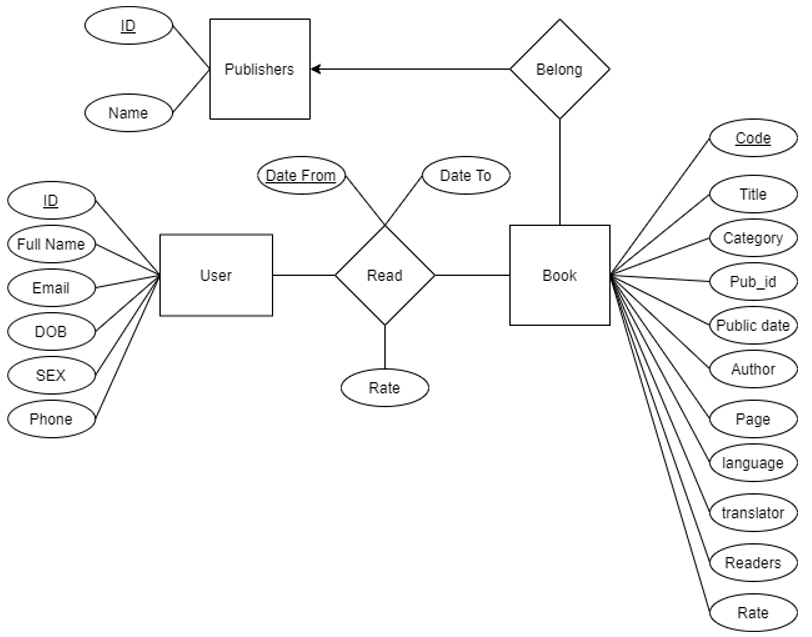
* For Publishers:

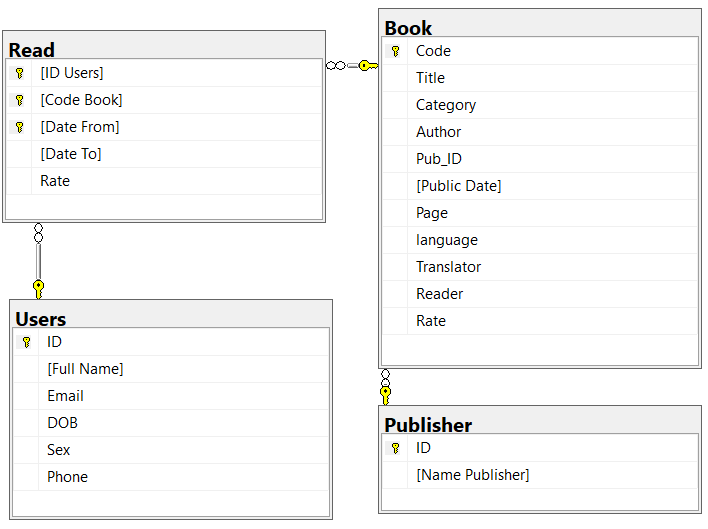
+ Manage Users’s ID : follow activate of users

+ Book: update/delete/insert books

+ Information: manage information of users

II. ERD.





* Create Table :

Create table Book(

Code char(13) Primary Key not null Check(Code like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

Title nvarchar(100) not null,

Category nvarchar(45) not null,

Author nvarchar(45) not null,

Pub\_ID int references Publisher(ID),

[Public Date] Date,

[Page] int not null,

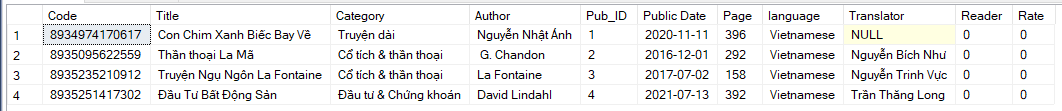
[language] nvarchar(45) not null,

Translator nvarchar(45),

Reader int,

Rate float

)

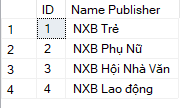


Create table Publisher(

ID int Primary Key not null,

[Name Publisher] nvarchar(45) not null,

)



create table Users(

ID int Primary Key not null,

[Full Name] nvarchar(45) not null,

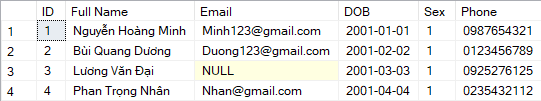
Email nvarchar(45),

DOB Date not null,

Sex char(1) not null check(Sex like '[0-1]'),

Phone char(10) not null Check(Phone like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

)



Create table [Read](

[ID Users] int references Users(ID) not null,

[Code Book] char(13) references Book(Code) not null,

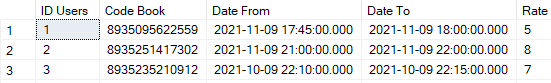
[Date From] Datetime not null,

[Date To] Datetime not null,

Rate float not null Check(Rate >=0 and Rate <=10),

Primary Key([ID Users], [Code Book],[Date From])

)



III. Proc, Trigger, Transaction.

1. Proc

-- **Main purpose**: Enter the code book will display the full names of those who have read this book .

Create proc show(

@code char(13)

)

As

Begin

Select distinct Users.[Full Name] From Users

Inner join "Read" on "Read".[ID Users] = Users.ID

Where "Read".[Code Book] = @code

End

exec show '8935095622559'

2. Trigger

-- **Main purpose**: The trigger will update the number of readers and the rate’s average of each book in Book table after insert value to Read table.

Create Trigger Insert\_Read

On [Read]

after Insert, Update

AS

BEGIN

declare @AvgOfRate float, @NumOfReader int, @code char(13)

SET @code = (Select i.[Code Book] FROM inserted i)

SELECT @NumOfReader = count(r.[ID Users]) FROM inserted i , [Read] r WHERE i.[Code Book] = r.[Code Book] Group By i.[Code Book]

SET @AvgOfRate = ((Select avg(r.[Rate]) FROM inserted i, [Read] r WHERE i.[Code Book] = r.[Code Book] Group By i.[Code Book]))

Update Book SET NoOfReader = @NumOfReader, AvgOfRate = @AvgOfRate WHERE Code = @code

END

-- Trigger 2

**Main purpose**: Trigger will display information of book was be deleted

Create trigger Del\_book

on book

after delete

as

begin

select \* from deleted

end

3. Transaction

-- **Main purpose**: Before delete any book in Book table, transaction will delete the records have same Code\_book in Read table, after that will delete that book in Book table, if have error will rollback

Create trigger delete\_book

on Book

instead of Delete

as

begin

declare @Code\_book char(13)

set @Code\_book = (select deleted.Code from deleted)

Begin transaction

delete from [Read]

where @Code\_book = [Code Book]

if @@ERROR <> 0

BEGIN

PRINT 'ERROR'

rollback

END

delete from Book

where @Code\_book = Code

if @@ERROR <> 0

BEGIN

PRINT 'ERROR'

rollback

END

Commit

end