NOVEMBER 9, 2021

ASSIGNMENT

DBI202 – DATABASE OF LIBRARY MANAGEMENT SYSTEM

STUDENT NAME: DANG VIET ANH | HE151359

STUDENT NAME: TRAN XUAN QUANG | HE153654

STUDENT NAME: DOAN HAI PHONG | HE151364

TEACHER: NGUYEN THI BICH THUY

TABLE OF CONTENTS

I) INTRODUCE THE PROBLEM

1) DESCRIBE THE 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.

2) GOALS

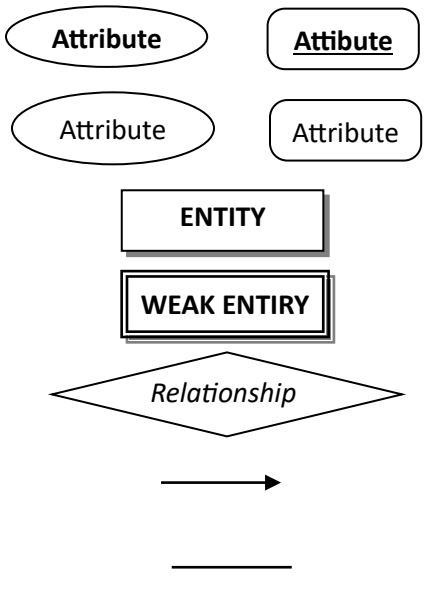
This database is written to solve the stages and procedures of the library by retrieving information on the computer without the need for manual work as before. This helps both library managers and readers easily control the information of books as well as the return of books in the library.

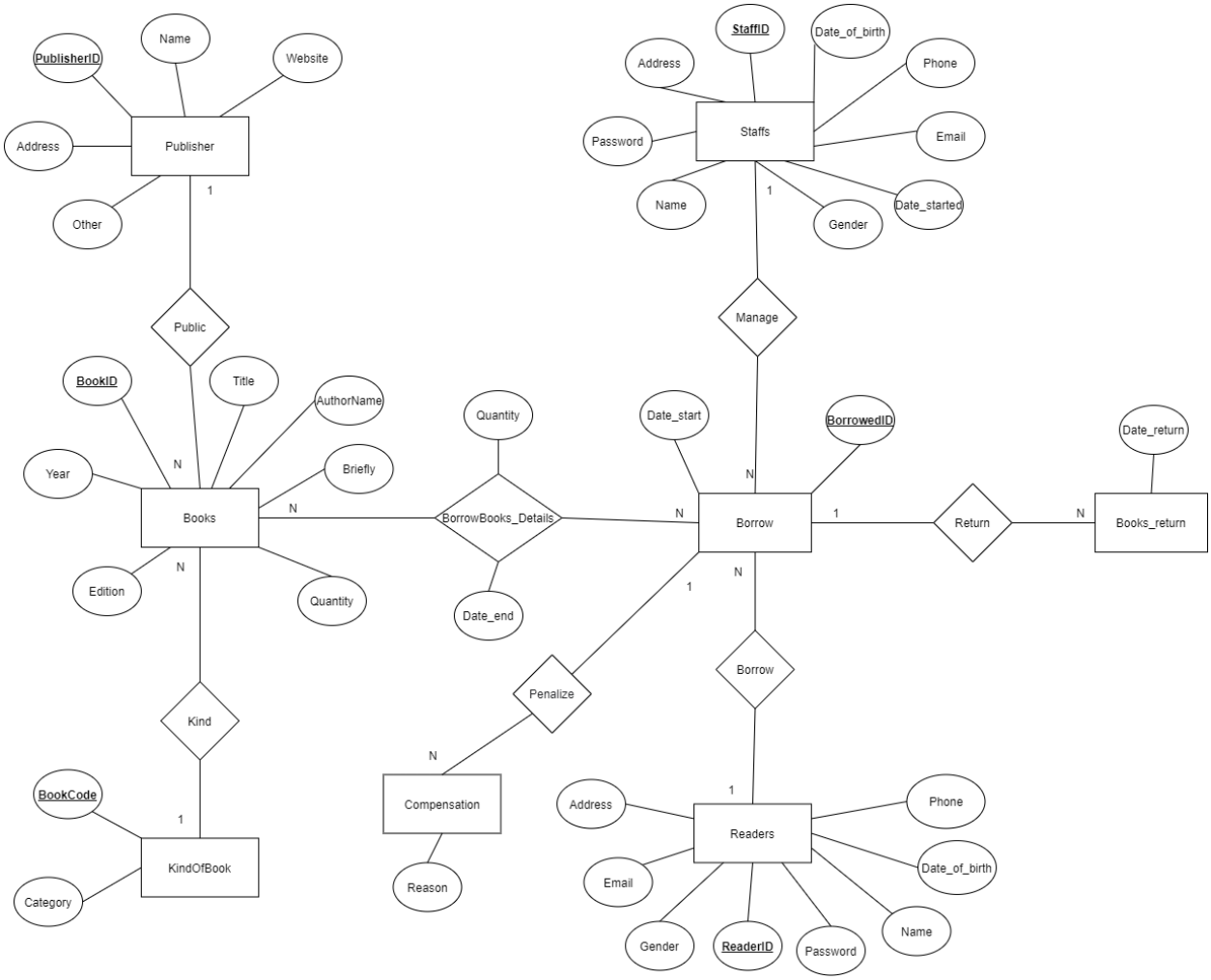
II) ENTITY – RELATIONSHIP – ERD

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





2) DIFINITION ENTITY – ATTRIBUTE

Base on the problem description and management objectives, we can present several entities and attributes of the entity as follow:

- **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.
- **Borrow**: BorrowedID, StaffID, ReaderID, Date_start.
- **BorrowBooks_details**: BorrowID,BookID, Quantity, Date_end.
- **Books_return**: BorrowID,Date_return.
- **Compensation**: BorrowedID, Reason.

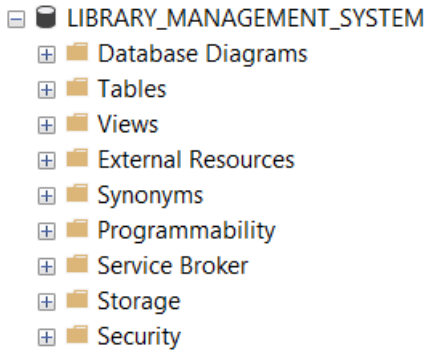
III) DATABASE AND ENTITY DIAGRAM

Just for example on some tables (other table are similar, you have to define all the tables in your database). Note: to run the query you have to define the table 1 first then go to the side tables much

A. CREATE DATABASE **LIBRARY_MANAGEMENT_SYSTEM**

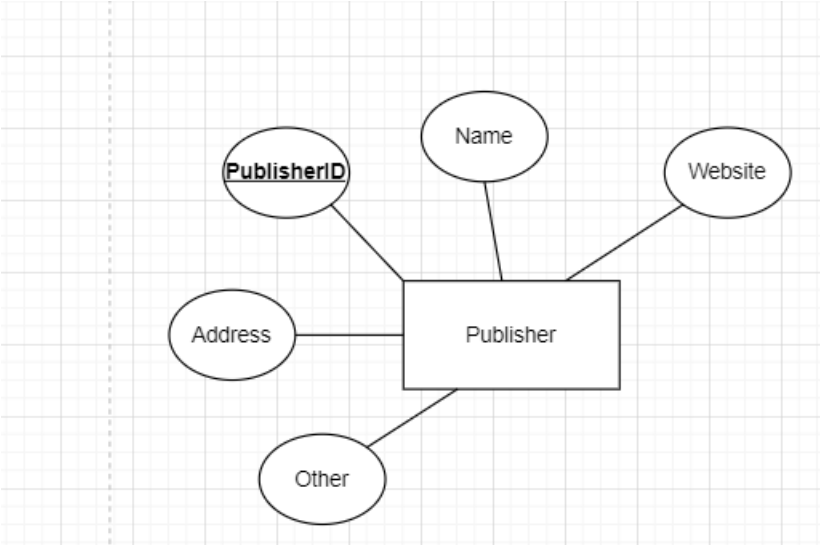
--create database

```
CREATE DATABASE LIBRARY_MANAGEMENT_SYSTEM
```



B. CREATE TABLE **PUBLISHER**

Entity - Attribute:



	Column Name	Data Type	Allow Nulls
🔑	PublisherID	char(50)	☐
	Name	nvarchar(200)	☐
	Address	nvarchar(200)	☐
	Website	char(100)	☑
	Other	nvarchar(200)	☑
			☐

Code:

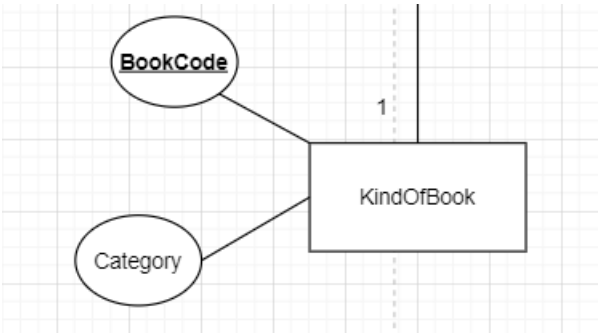
```
--create table Pulisher
	CREATE TABLE Publisher(
	PublisherID char(50) NOT NULL PRIMARY KEY,
	Name nvarchar(200) NOT NULL,
	Address nvarchar(200) NOT NULL,
	Website char(100),
	Other nvarchar(200))
```

Example:

	PublisherID	Name	Address	Website	Other
1	AEEBD	Phụ nữ	39 Hàng Chuối, Hà Nội	PhuNu.vn	Chuyên về các đầu sách của phái đẹp
2	AEPHB	Thanh niên	Số 64 Bà Triệu - Hoàn Kiếm - Hà Nội	NXBThanhNien.vn	NULL
3	AJZZB	Y học	352 Đội Cấn, Ba Đình, Hà Nội	Yhoc.vn	Chuyên về các đầu sách phục vụ ngành y
4	AZCWE	Lao động - Xã hội	Số 36, Ngõ hoà bình 4 - Minh khai - Hai Bà Trưng - Hà ...	NULL	NULL
5	BVROO	Âm nhạc	61 Lý Thái Tổ - Hoàn Kiếm - Hà Nội	NULL	Các loại sách âm nhạc giải trí

C. CREATE TABLE **KINDOFBOOK**

Entity - Attribute:



	Column Name	Data Type	Allow Nulls
🔑	BookCode	char(50)	☐
	Category	nvarchar(200)	☐
			☐

Code:

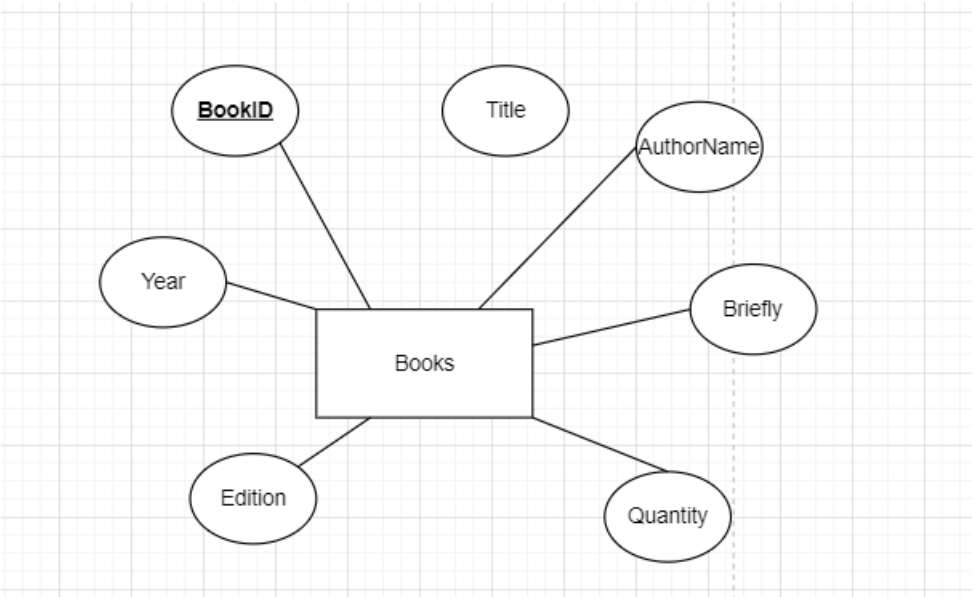
```
--create table KindOfBook
CREATE TABLE KindOfBook(
BookCode char(50) NOT NULL PRIMARY KEY,
Category nvarchar(200) NOT NULL)
```

Example:

	BookCode	Category
1	AT-NA	Ẩm Thực - Nấu Ăn
2	CNTT	Công Nghệ Thông Tin
3	GT	Giải Trí
4	HT	Tài Liệu Học Tập - Giáo Trình
5	KH-KT	Khoa Học - Kỹ Thuật
6	KT-QL	Kinh Tế - Quản Lý
7	KT-XD	Kiến Trúc - Xây Dựng
8	LS	Lịch Sử

D. CREATE TABLE **BOOKS**

Entity - Attribute:



	Column Name	Data Type	Allow Nulls
🔑	BookID	char(50)	☐
	Title	nvarchar(100)	☐
	AuthorName	nvarchar(100)	☑
	PublisherID	char(50)	☐
	BookCode	char(50)	☐
	Year	int	☑
	Edition	int	☑
	Quantity	int	☐
	Briefly	nvarchar(1000)	☑
			☐

Code:

```
--create table Books
create table Books(
BookID char(50) NOT NULL PRIMARY KEY,
Title nvarchar(100) NOT NULL,
AuthorName nvarchar(100),
```

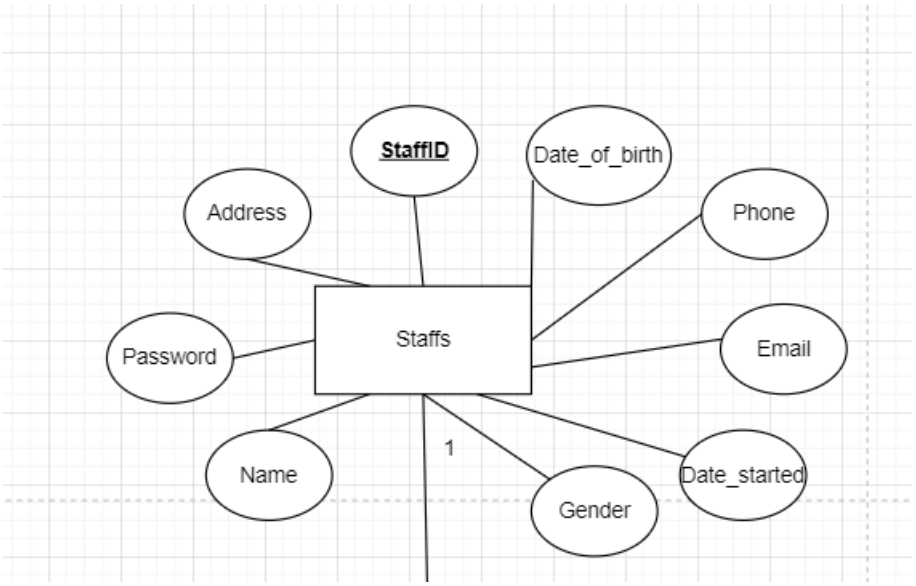
```
PublisherID char(50) NOT NULL,
BookCode char(50) NOT NULL,
Year int check (Year <= YEAR(getDate())),
Edition int,
Quantity int NOT NULL,
Briefly nvarchar(1000)
constraint fk_Books_KindOfBook FOREIGN KEY(BookCode) references KindOfBook(BookCode),
constraint fk_Books_Publisher FOREIGN KEY(PublisherID) references Publisher(PublisherID)
 )
```

Example:

	BookID	Title	AuthorName	PublisherID	BookCode	Year	Edition	Quantity	Briefly
1	054JD	Vật Liệu Và Thiết Bị NANO	Trương Văn Tân	AEPHB	CNTT	2018	5	20	Hình dung và nắm bắt khá đầy đủ về một phạm trù k...
2	0EUWJ	Châu Âu Có Gì Lạ Không Em	Misa Gjone	YVWTZ	NN	2019	5	19	Cuốn sách là những ghi chép và mô tả của tác giả về...
3	2TIVX	Glory over Everything: Beyond The Kitchen House	Bottom Hust	UOIDA	GT	2016	5	25	NULL
4	3EJ3S	Kinh điển về khởi nghiệp	Dan Senor	YWDLZ	KT-QL	2012	14	30	Dù khởi nghiệp kinh doanh là con đường mà bạn đã...
5	45FHD	Bước đi ngẫu nhiên trên phố Wall	Burton G.Malkiel	DZXDT	TC-CK	2018	7	10	Sơ lược về chứng khoán
6	45LPK	Lịch Sử Chiến Tranh	John Keegan	ELBAJ	LS	2018	10	22	Chiến tranh là văn minh hay dã man? Trả lời câu hỏi...

E. CREATE TABLE **STAFFS**

Entity - Attribute:



Code:

	Column Name	Data Type	Allow Nulls
🔑	StaffID	char(50)	☐
	Name	nvarchar(100)	☐
	Address	nvarchar(200)	☑
	Date_of_birth	date	☐
	Gender	bit	☐
	Phone	char(50)	☑
	Email	char(100)	☑
	Date_started	date	☐
	Password	char(50)	☐
			☐

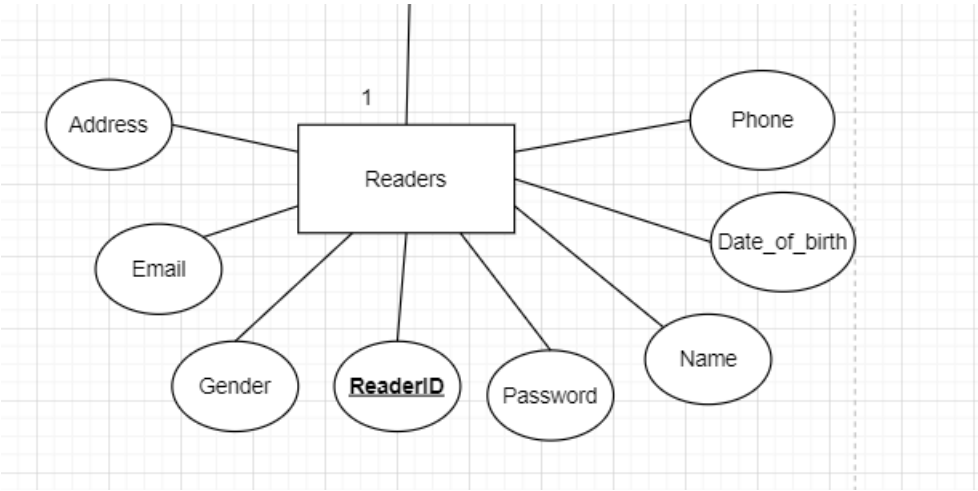
```
--create table Staffs
CREATE TABLE Staffs(
StaffID char(50) NOT NULL PRIMARY KEY CHECK (StaffID like 'FU[0-9][0-9][0-9][0-9][0-9][0-9]'),
Name nvarchar(100) NOT NULL,
Address nvarchar(200),
Date_of_birth date NOT NULL check (Date_of_birth < getDate()),
Gender bit NOT NULL check (Gender like 0 or Gender like 1),
Phone char(50) UNIQUE CHECK (Phone like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
Email char(100),
Date_started date NOT NULL,
Password char(50) NOT NULL
 )
```

Example:

	StaffID	Name	Address	Date_of_birth	Gender	Phone	Email	Date_started	Password
1	FU111254	Nguyễn Vũ Hoàng	15, Thái Thịnh, Thịnh Quang, Đống Đa, Hà Nội	1988-06-11	1	0354985818	thanhtan12@gmail.com	2018-11-21	123456789
2	FU130487	Nguyễn Thế Nguyên	201, Thái Thịnh, Trung Liệt, Đống Đa, Hà Nội	1990-06-25	1	0865074574	baongoc32@gmail.com	2020-11-11	123456789
3	FU133558	Trần Văn Quý	123, Tam Khương, Khương Thượng, Đống Đa, Hà Nội	1980-10-24	1	0866811237	tuankhang19@gmail.com	2017-07-11	123456789
4	FU133578	Nguyễn Thành Đạt	103, Lê Trọng Tấn, Khương Trung, Thanh Xuân, Hà Nội	1980-06-20	1	0359757368	tonuwxa@gmail.com	2018-06-23	123456789
5	FU133654	Đặng Mạnh Cường	103, đường Tam Khương, Khương Thượng, Đống Đa, ...	1980-05-25	0	0373243088	ngocthien65@gmail.com	2018-07-02	123456789
6	FU140396	Đoàn Hải Phong	321, Pháo Đài Láng, Láng Thượng, Đống Đa, Hà Nội	1988-05-10	0	0367910186	tuanvu14@gmail.com	2018-10-05	123456789

F. CREATE TABLE **READERS**

Entity - Attribute:



	Column Name	Data Type	Allow Nulls
🔑	ReaderID	char(50)	☐
	Name	nvarchar(100)	☐
	Address	nvarchar(200)	☑
	Date_of_birth	date	☐
	Email	char(100)	☑
	Gender	bit	☐
	Password	char(50)	☐
			☐

Code:

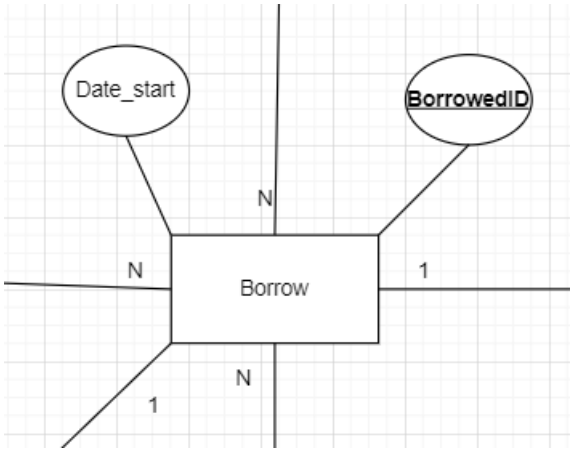
```
--create table Readers
create table Readers(
ReaderID char(50) NOT NULL PRIMARY KEY CHECK (ReaderID like '[A-Z][A-Z][0-9][0-9][0-9][0-9][0-9][0-9]'),
Name nvarchar(100) NOT NULL,
Address nvarchar(200),
Date_of_birth date NOT NULL check (Date_of_birth < getDate()),
Email char(100),
Gender bit NOT NULL check (Gender like 0 or Gender like 1),
Password char(50) NOT NULL
 )
```

Example:

	ReaderID	Name	Address	Date_of_birth	Email	Gender	Password
1	HE151332	Nguyễn Minh Vương	Hưng Yên	2001-07-04	vuongnmHE151332@fpt.edu.vn	1	123456789
2	HE151333	Nguyễn Đức Anh	Phú Thọ	2001-08-05	anhndHE151333@fpt.edu.vn	1	123456789
3	HE151334	Vũ Tiến Đạt	Nam Định	2001-12-14	dattvHE151334@fpt.edu.vn	1	123456789
4	HE151335	Nguyễn Duy Đạt	Hà Nội	2001-01-01	datndHE151335@fpt.edu.vn	1	123456789
5	HE151336	Lê Hữu Phúc	Hà Nội	2001-03-02	HE151336@fpt.edu.vn	1	123456789
6	HE151337	Ngô Quốc Tuấn	Hà Nội	2001-05-04	tuannqHE151336@fpt.edu.vn	1	123456789

G. CREATE TABLE **BORROW**

Entity - Attribute:



	Column Name	Data Type	Allow Nulls
🔑	BorrowID	char(50)	☐
	StaffID	char(50)	☐
	ReaderID	char(50)	☐
	Date_start	date	☐
			☐

Code:

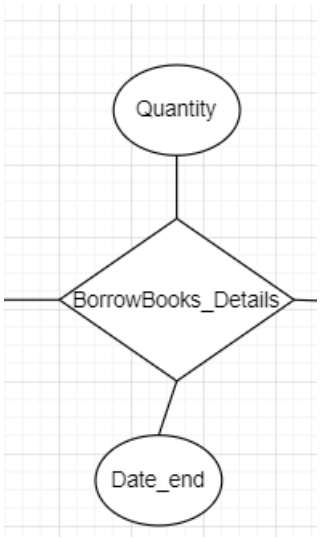
```
--create table Borrow
create table Borrow(
BorrowID char(50) NOT NULL PRIMARY KEY,
StaffID char(50) NOT NULL,
ReaderID char(50) NOT NULL,
Date_start date NOT NULL,
constraint fk_Borrow_Readers FOREIGN KEY(ReaderID) references Readers(ReaderID),
constraint fk_Borrow_Staffs FOREIGN KEY(StaffID) references Staffs(StaffID))
```

Example:

	BorrowID	StaffID	ReaderID	Date_start
1	1	FU143548	HE151332	2021-10-24
2	10	FU140396	HE151342	2021-10-07
3	11	FU143548	HE151343	2021-10-06
4	12	FU133558	HE151344	2021-10-05
5	13	FU143685	HE151345	2021-10-04
6	14	FU111254	HE151346	2021-10-03

H. CREATE TABLE **BORROWBOOKS_DETAILS**

Entity - Attribute:



	Column Name	Data Type	Allow Nulls
🔑	BorrowID	char(50)	☐
🔑	BookID	char(50)	☐
	Quantity	int	☐
	Date_end	date	☐
			☐

Code:

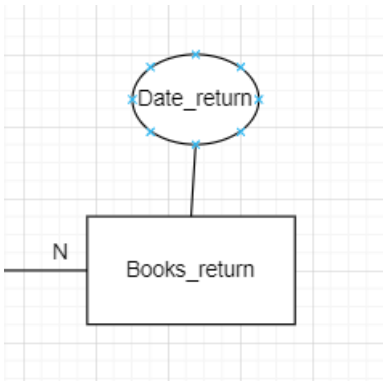
```
--create table BorrowBooks_details
CREATE TABLE BorrowBooks_details(
BorrowID char(50) NOT NULL,
BookID char(50) NOT NULL,
Quantity int NOT NULL,
Date_end date NOT NULL,
constraint fk_BorrowBooks_details_Borrow FOREIGN KEY(BorrowID) references
Borrow(BorrowID),
constraint fk_BorrowBooks_details_Books FOREIGN KEY(BookID) references Books(BookID)
primary key(BorrowID,BookID)
)
```

Example:

	BorrowID	BookID	Quantity	Date_end
1	1	QPLDS	49	2021-11-24
2	1	PLSJD	1	2021-11-24
3	1	AKDPS	1	2021-11-24
4	2	054JD	2	2021-11-15
5	2	QPLDS	3	2021-11-15
6	3	94JDX	1	2021-11-14
7	3	ALSKO	2	2021-11-14

I. CREATE TABLE **BOOKS_RETURN**

Entity - Attribute:



	Column Name	Data Type	Allow Nulls
▶	BorrowID	char(50)	☐
	Date_return	date	☐
			☐

Code:

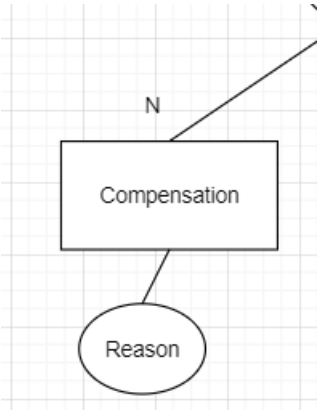
```
--create table Books_return
CREATE TABLE Books_return(
BorrowID char(50) NOT NULL,
Date_return date NOT NULL,
constraint fk_Books_return_Borrow FOREIGN KEY(BorrowID) references Borrow(BorrowID))
```

Example:

	BorrowID	Date_return
1	1	2021-11-24
2	2	2021-11-13
3	3	2021-11-22
4	4	2021-11-05
5	5	2021-11-11
6	6	2021-11-21
7	7	2021-11-22
8	8	2021-11-23
9	9	2021-11-07
10	10	2021-11-06
11	11	2021-11-05

J. CREATE TABLE **COMPENSATION**

Entity - Attribute:



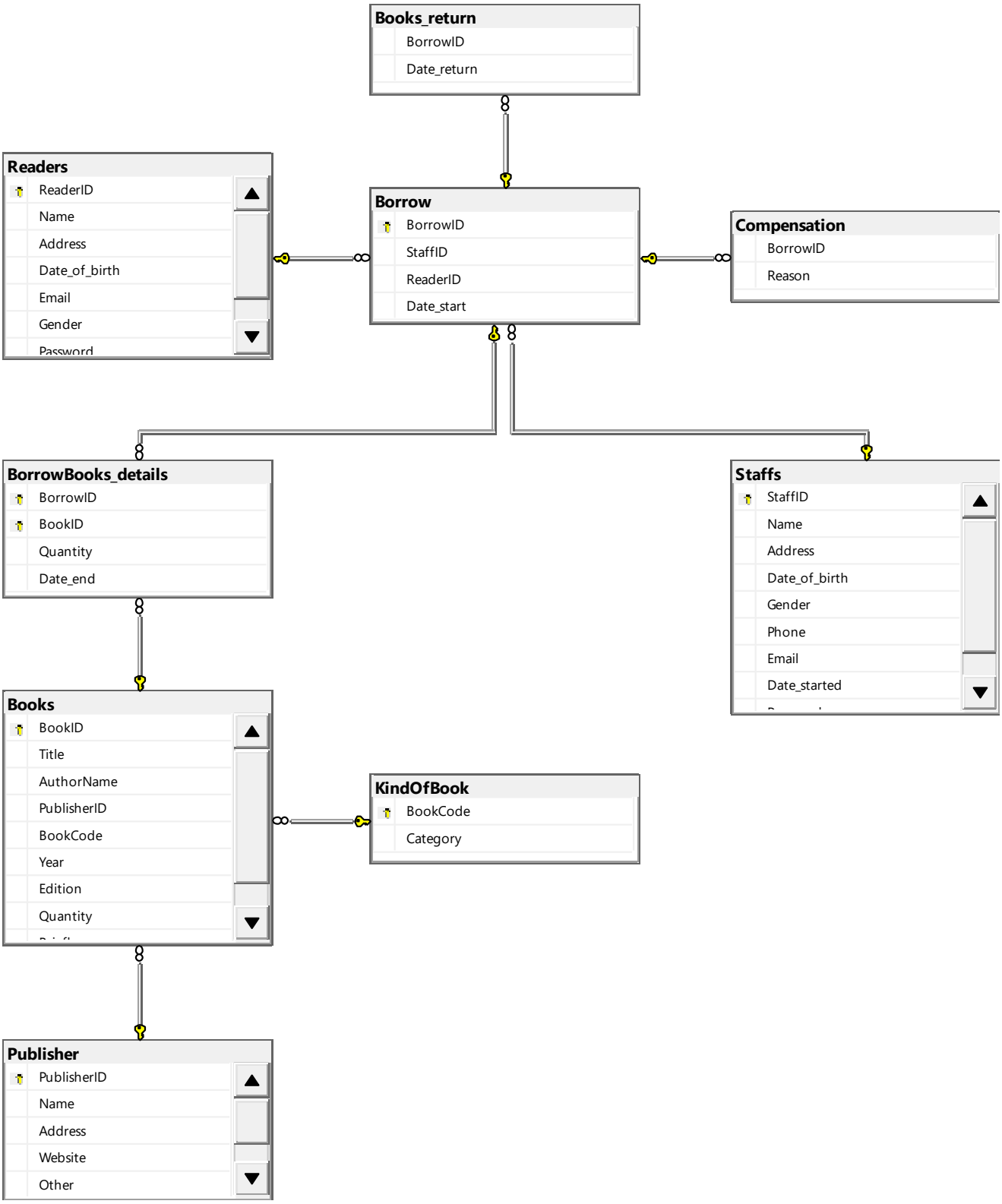
	Column Name	Data Type	Allow Nulls
▶	BorrowID	char(50)	☐
	Reason	nvarchar(200)	☐
			☐

Code:

```
--create table Compensation
create table Compensation(
BorrowID char(50) NOT NULL,
Reason nvarchar(200) NOT NULL,
constraint fk_PHAT_MUONSACH FOREIGN KEY(BorrowID) references Borrow(BorrowID))
```

Example:

	BorrowID	Reason
	3 ...	Bị nộp muộn 8 ngày
	6 ...	Bị nộp muộn 10 ngày
	7 ...	Bị nộp muộn 12 ngày
	8 ...	Bị nộp muộn 14 ngày
▶*	*NULL*	*NULL*


Books_return
BorrowID
Date_return
Readers
ReaderID
Name
Address
Date_of_birth
Email
Gender
Password
Borrow
BorrowID
StaffID
ReaderID
Date_start
Compensation
BorrowID
Reason
BorrowBooks_details
BorrowID
BookID
Quantity
Date_end
Staffs
StaffID
Name
Address
Date_of_birth
Gender
Phone
Email
Date_started
Books
BookID
Title
AuthorName
PublisherID
BookCode
Year
Edition
Quantity
KindOfBook
BookCode
Category
Publisher
PublisherID
Name
Address
Website
Other

V. SQL COMMAND

I using Microsoft SQL Server 2019, this server build intelligent, mission-critical applications using a scalable, hybrid database platform that has everything built in—from in-memory performance and advanced security to in-database analytics.

A. QUERY USING ORDER BY

Code:

```
SELECT * FROM Staffs s
ORDER BY s.Name
```

Result:

	StaffID	Name	Address	Date_of_birth	Gender	Phone	Email	Date_started	Password
1	FU133654	Đặng Mạnh Cường	103, đường Tam Khương, Khương Thượng, Đống Đa, ...	1980-05-25	0	0373243088	ngocthien65@gmail.com	2018-07-02	123456789
2	FU140396	Đoàn Hải Phong	321, Pháo Đài Láng, Láng Thượng, Đống Đa, Hà Nội	1988-05-10	0	0367910186	tuanvu14@gmail.com	2018-10-05	123456789
3	FU143685	Lê Văn Lâm	169, Tây Sơn, Quang Trung, Đống Đa, Hà Nội	1987-09-26	1	0342666531	tamtang20@gmail.com	2016-09-10	123456789
4	FU133578	Nguyễn Thành Đạt	103, Lê Trọng Tấn, Khương Trung, Thanh Xuân, Hà Nội	1980-06-20	1	0359757368	tonuwxa@gmail.com	2018-06-23	123456789
5	FU130487	Nguyễn Thế Nguyên	201, Thái Thịnh, Trung Liệt, Đống Đa, Hà Nội	1990-06-25	1	0865074574	baongoc32@gmail.com	2020-11-11	123456789
6	FU143548	Nguyễn Văn A	368, Âu Cơ, Hạ Hòa, Hạ Hòa, Phú Thọ	1980-06-26	1	0333204166	thudong11@gmail.com	2018-07-21	123456789
7	FU111254	Nguyễn Vũ Hoàng	15, Thái Thịnh, Thịnh Quang, Đống Đa, Hà Nội	1988-06-11	1	0354985818	thanhtan12@gmail.com	2018-11-21	123456789
8	FU142708	Thân Văn Lâm	183, Hoàng Văn Thái, Khương Mai, Thanh Xuân, Hà Nội	1980-02-26	1	0348387386	nhuvan19@gmail.com	2018-05-21	123456789
9	FU153204	Tiết Nguyên Thành	184, Trường Chinh, Khương Thượng, Đống Đa, Hà Nội	1989-06-20	1	0335151714	trandan55@gmail.com	2017-07-20	123456789
10	FU133558	Trần Văn Quý	123, Tam Khương, Khương Thượng, Đống Đa, Hà Nội	1980-10-24	1	0866811237	tuankhang19@gmail.com	2017-07-11	123456789

We use query containing ORDER BY to sort the list ascending or descending by the values of a domain. SELECT * FROM Staffs command give us all record in Staffs table and then sort the records ascending by name

B. QUERY USING INNER JOIN

Code:

```
SELECT * FROM Books b
inner join Publisher p
ON b.PublisherID = p.PublisherID
WHERE b.Year = 2018
```

Result:

	BookID	Title	Year	Website
1	054JD	Vật Liệu Và Thiết Bị NANO	2018	NXBThanhNien.vn
2	45FHD	Bước đi ngẫu nhiên trên phố Wall	2018	NULL
3	45LPK	Lịch Sử Chiến Tranh	2018	NULL
4	83HDY	Nụ Cười Việt Nam	2018	NULL
5	842HD	Lê La Quà Vặt	2018	NULL
6	94JDX	Huyền Thoại Nhạc Pop	2018	NULL
7	94JFH	Giáo Trình Kỹ Thuật Audio Và Video	2018	NULL
8	ALSKO	Lập Và Giải Từ Vi Đấu Số	2018	PhuNu.vn
9	ASH32	Giải Pháp Cho Internet Vạn Vật	2018	NXBThanhNien.vn
10	LFKDS	Rock Hà Nội, Bolero Sài Gòn	2018	NULL
11	LPSOR	10 Đồng Tiền Thay Đổi Thế Giới	2018	NULL
12	OWIED	Câu Chuyện Nghệ Thuật	2018	NULL
13	QPLDS	Trái Tim Của Bé	2018	NULL
14	SLKN3	Mô Hình Điều Khiển Từ Xa	2018	NULL

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

C. QUERY USING AGGREGATE FUNCTIONS

Code:

```
SELECT COUNT(B.BookID) FROM Books B, Publisher P
WHERE B.PublisherID = P.PublisherID
AND P.Name = N'Kim Đồng'
```

Result:

	(No column name)
1	7

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

Code:

```
SELECT TOP(5) * FROM Books
ORDER BY Quantity DESC
```

Result:

	BookID	Title	AuthorName	PublisherID	BookCode	Year	Edition
1	77SQ4	Hà Nội ngàn năm kí ức	Phương Chi	UOIDA	VH-NT	2021	3
2	G2IKF	Cuốn Sách Về Quyền Lực - Nó Là Cái Gì, Ai Có Nó,...	Thủy Dung	UOIDA	TL	2020	2
3	GCWWS	Robert Kirkman	Tony Moore, Charlie Adlard, Cliff Rathburn	UTSRP	GT	2004	5
4	POR0I	Sổ Tay Kỹ Thuật Thi Công Nhà Ở Gia Đình	Nguyễn Bá Đô	AZCWE	KT-XD	2011	2
5	4U7X8	Prophet	Lida Clause	UOIDA	GT	1992	4

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

D. QUERY USING THE GROUP BY AND HAVING CLAUSES

Code:

```
SELECT S.StaffID, S.Name, COUNT(B.BorrowID) AS TOTALMANAGE FROM Staffs S, Borrow B
WHERE S.StaffID = B.StaffID
GROUP BY S.StaffID, S.Name
HAVING COUNT(B.BorrowID) > 3
```

Result:

	StaffID	Name	TOTALMANAGE
1	FU133558	Trần Văn Quý	4
2	FU143548	Nguyễn Văn A	4
3	FU143685	Lê Văn Lâm	4

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

E. QUERY THAT USES A SUB-QUERY AS A RELATION

Code1:

```
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)
```

Result1:

	BookID	Title	TotalBorrows
1	AKDPS	Tử Vi Đẩu Số Phi Tinh	4
2	HFJ3D	Nhà đầu tư thông minh	4
3	KODJK	Lược khảo văn học	4
4	WUEJF	Nghịch Lý Của Sự Lựa Chọn	4

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

Code2:

```
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)
```

Result2:

Results | Messages

	ReaderID	Name	NumberOfBorrowedBooks
1	HE151332	Nguyễn Minh Vương	51

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

Code:

```
SELECT *
FROM Books
WHERE Title like N'%Trí Tuệ%' and Quantity > 10
```

Result:

Results | Messages

	BookID	Title	AuthorName	PublisherID	BookCode	Year	Edition	Quantity	Briefly
1	9RUDH	Trí Tuệ Dân Tộc Úc	Kiến Văn , Gia Khang	VJTYN	HT	2020	10	15	Những đặc trưng khái quát về một đất nước, một d...

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

G. STORE PROCEDURE

Code:

```
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
```

Result:

Results | Messages

	Quantity
1	20

We use procedure to check the quantity of books for which the book's ID is entered by the librarian or user.

H. TRIGGER

Code1:

```
CREATE TRIGGER checkCompensation
ON Books_return AFTER INSERT
AS
BEGIN
	DECLARE @borrowID CHAR(50);
	DECLARE @date_end INT
	SELECT @borrowID = i.BorrowID, @date_end = DATEDIFF(day,bd.Date_end,i.Date_return)
	FROM inserted i,
	(SELECT DISTINCT bd.BorrowID, bd.Date_end FROM BorrowBooks_details bd) AS bd
	WHERE bd.BorrowID = i.BorrowID
	IF(@date_end >0)
```

```
	BEGIN
		INSERT INTO Compensation(BorrowID,Reason) VALUES
		(@borrowID, N'Bị nộp muộn ' + CONVERT(NVARCHAR(200),@date_end) + N' ngày')
	END
END
```

Result1:

	BorrowID	Reason
▶		Bị nộp muộn 8 ngày
	6 ...	Bị nộp muộn 10 ngày
	7 ...	Bị nộp muộn 12 ngày
	8 ...	Bị nộp muộn 14 ngày
*	*NULL*	*NULL*

Add overdue loan slip to the table compensation

Code2:

```
CREATE TRIGGER checkRemainQuanity
ON BorrowBooks_details AFTER INSERT
AS
BEGIN
	DECLARE @quantity INT;
	SELECT @quantity = (SELECT b.Quantity - (SELECT sum(b.Quantity) FROM inserted i,
	BorrowBooks_details b
	WHERE i.BookID = b.BookID)
	FROM Books b, inserted i WHERE b.BookID = i.BookID)
	IF(@quantity<0)
	BEGIN
		PRINT 'Not enough quantity'
		DELETE FROM BorrowBooks_details
		where Quantity = (SELECT i.Quantity FROM inserted i) and
		BorrowID = (SELECT i.BorrowID FROM inserted i)
		and BookID = (SELECT i.BookID FROM inserted i)
	END
END
```

Check to see if the quantity of books is enough for the borrower

Result2:

Messages

```
Not enough quantity

(2 rows affected)

(1 row affected)
```

THE END