

Project Design Report

[Library management database system]

[Amigo]

[11/21/2022]

Peiyu Li

Preethi Reddy Tera

Yagnashree Velanki

Introduction

The library is one of the most essential parts of each university, it provides students with various sources. In this project, we covered a library mini world that stores the data of the person entity with different roles and the data of facilities in the library. In our work, we build different relationships (borrow, return, reserves, purchases) between the person entity and different entities of facilities (reading materials, study rooms, labs, printers, quick shop) in the library. In particular, the person entity is specialized into different subclass entities with different roles. Different roles have different privileges. For example, anyone can borrow books from the library, but there are study rooms only designed for students to reserve to schedule group meetings for their final projects; and lecture halls can only be reserves by faculty to give public talks.

Some of the features or aspects that we have added in our design are:

1. There is a settlement section, if the borrowed books or gadgets get damaged then a penalty will be issued to the user and the amount is reflected in the penalty section.
2. Membership is created for the guest who are not a member of a university and willing to use library resources. Guests need to buy memberships to get access to the resources in the library.
3. We have created a separate entity called WIFI in which each member that comes into a library should enter that network using the password to log in to access library resources.
4. We created a quick shop entity called Quickshop in which each member can purchase products they need.
5. All the employees working in the library cannot modify the details of books or students. Only a few employees working in admin positions have the authority to alter the details.

Requirements Analysis (functional requirements)

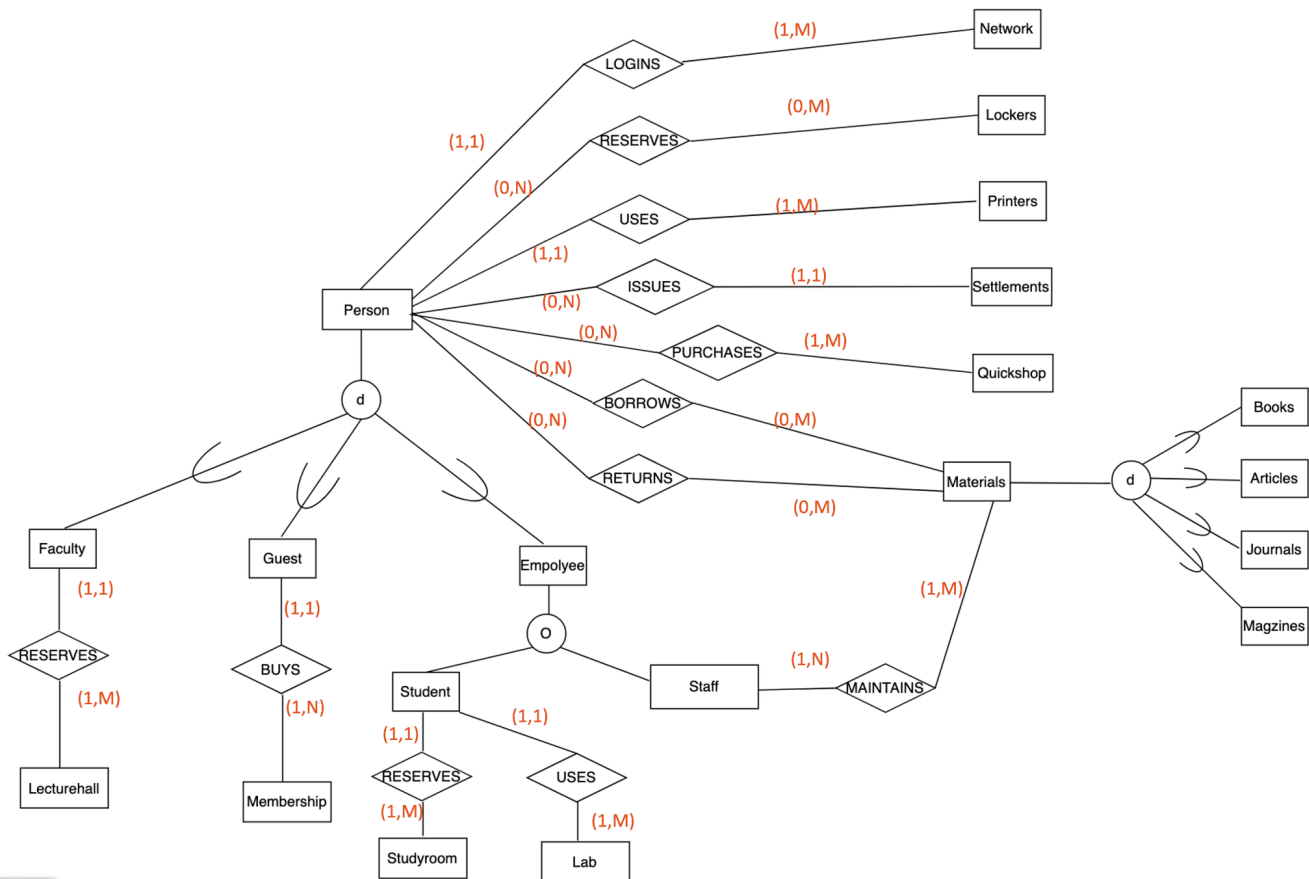
Currently we are focusing on the college/university library management system. Our requirements are

1. Users can check the availability of books/resource by the title/author of books
2. Library staff can query the information of books that have been borrowed and the information of the person that borrowed the books
3. Students can query the availability of study rooms
4. Professors can query the availability of lecture halls
5. The information of books should be updated, for example, if the books have been returned, the availability of the books for other students should be updated
6. The availability of study rooms should be updated in time, after a reservation is made by a student, the availability of that study room that was reserved by the student should be set to 'False'; if the students leave, the availability of that study room should be set to 'True'
7. The availability of lecture halls should be updated in time, after a reservation is made by a faculty, the availability of that lecture hall that was reserved by the faculty should be set to 'False'; if the faculty leaves, the availability of that lecture hall should be set to 'True'

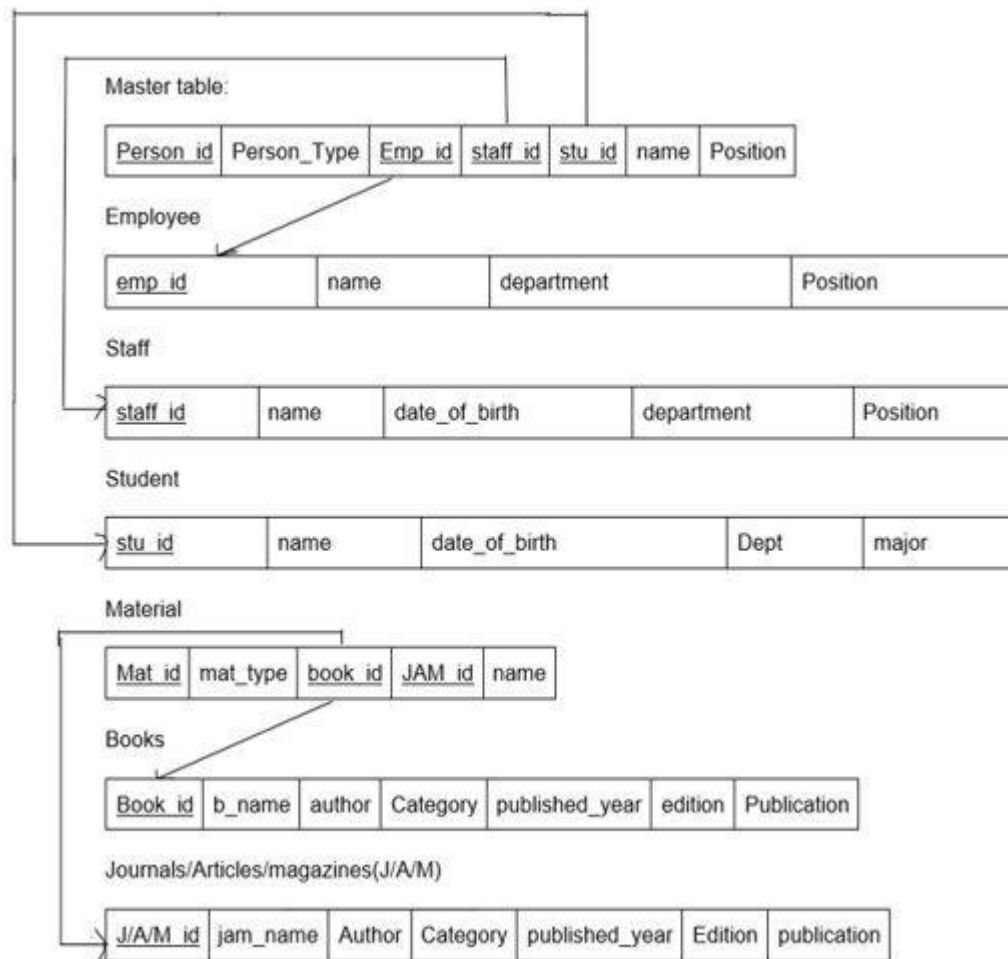
additional contains:

- To avoid conflicts, resources like study rooms, lockers, and lecture halls need to be reserved before use. While making a reservation, the start time and the end time should be stored
- Students/staff having student/staff IDs should be members of the library and do not need any other memberships to use the library resources. But people from outside of the university can access the library if they buy a membership card from the library.
- Penalty will be issued if the borrowed book is damaged or not returned on time
- Users can only borrow up to 5 books at a time. If they try to borrow a new book after borrowing 5 books a trigger is issued saying the maximum limit reached.

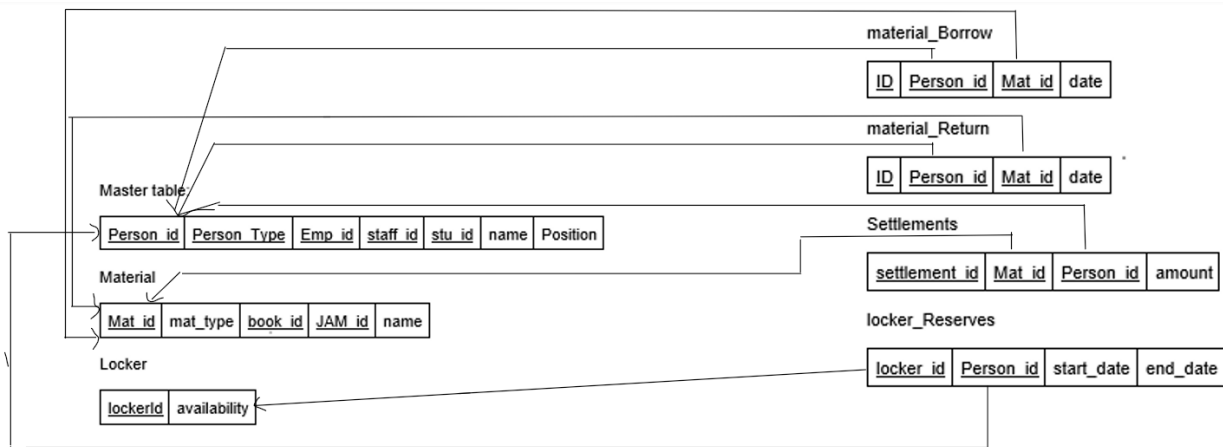
EER model



Relational model



For the Above relation Apart from the Master tables i.e person table and material table everything is in #NF naturally. The Master tables are created for easy access of relations between the tables and to get the general data of a person or material without going into the primary tables. We have added the foreign key constraints as the primary key is auto generated in the master tables and the column details are auto generated when the primary tables are updated using triggers. So in case if trigger are removed then there will be integrity constraints on the table when the relations are given. so to avoid such scenarios we have taken the foreign key constraints on the master tables.



Above tables are in the 3NF form. for Tables material_borrow, material_return, Settlements Id, person_id, mat_id combinedly forms the primary key as we cannot describe which person used which material without person_id and mat_id. Hence they satisfy the 3NF as they will not make any sense if we split them further.

Reservation between faculty entity and lecture hall entity

This reservation relation already follows at least 3NF, firstly, the domains of all attributes include only atomic values which is in 1NG; secondly, every nonkey attribute is fully functionally dependent on a primary key; finally, every nonkey attribute is nontransitively dependent on the primary key.

Faculty → Relation name

<u>faculty_Id</u>	name	Department
2200001	Kevin	Math
2200003	Jack	CS

Attributes: faculty_Id, name, Department
Tuples: (2200001, Kevin, Math), (2200003, Jack, CS)

Lecturehall Reservation

<u>faculty_Id</u>	<u>hall_Id</u>	<u>Date</u>	startTime	endTime
2200001	101	2022-01-01	10:00:00	11:00:00
2200003	201	2022-09-30	14:00:00	15:00:00

Lecture Hall → Relation name

<u>hall_Id</u>	Capacity	Occupied
101	30	1
201	100	0

Attributes: hall_Id, Capacity, Occupied
Tuples: (101, 30, 1), (201, 100, 0)

Reservation between student entity and study room entity

This reservation relation already follows at least 3NF, firstly, the domains of all attributes include only atomic values which is in 1NG; secondly, every nonkey attribute is fully functionally dependent on a primary key; finally, every nonkey attribute is nontransitively dependent on the primary key.

Student → Relation name

stu_Id	name	DOB	department	position	major
S01	Jake	1998-10-07	CS	graduate	CS
S02	Tony	1976-01-07	CS	graduate	DS

→ Attributes

} Tuples

Studyroom Reservation

stu_Id	roomId	re_Date	startTime	endTime
S01	1001	2022-11-23	10:30:00	12:30:00
S02	1002	2022-11-24	08:30:00	10:30:00

Stdyroom → Relation name

<u>roomId</u>	Capacity	Occupied
1001	3	0
1002	3	0

→ Attributes

} Tuples

Mapping

Master table:

<u>Person_id</u>	<u>Person_Type</u>	<u>Emp_id</u>	<u>staff_id</u>	<u>stu_id</u>	name	Position
------------------	--------------------	---------------	-----------------	---------------	------	----------

Employee

<u>emp_id</u>	name	department	Position
---------------	------	------------	----------

Staff

<u>staff_id</u>	name	date_of_birth	department	Position
-----------------	------	---------------	------------	----------

Student

<u>stu_id</u>	name	date_of_birth	Dept	major
---------------	------	---------------	------	-------

Material

<u>Mat_id</u>	mat_type	<u>book_id</u>	<u>JAM_id</u>	name
---------------	----------	----------------	---------------	------

Books

<u>Book_id</u>	b_name	author	Category	published_year	edition	Publication
----------------	--------	--------	----------	----------------	---------	-------------

Journals/Articles/magazines(J/A/M)

<u>J/A/M_id</u>	jam_name	Author	Category	published_year	Edition	publication
-----------------	----------	--------	----------	----------------	---------	-------------

Studyroom

<u>roomId</u>	occupied	capacity
---------------	----------	----------

Study Room Reservation

stuld	roomId	Date	startTime	endTime
-------	--------	------	-----------	---------

Faculty

<u>facultyId</u>	name	phone
------------------	------	-------

Lecture Hall Reservation

<u>facultyId</u>	<u>hallId</u>	<u>Date</u>	startTime	endTime
------------------	---------------	-------------	-----------	---------

Lecturehall

<u>hallId</u>	occupied	capacity
---------------	----------	----------

Guest

<u>memberId</u>	name	phone
-----------------	------	-------

Membership

<u>id</u>	memberId	startDate	EndDate
-----------	----------	-----------	---------

Lab

<u>labId</u>	type	capacity
--------------	------	----------

Printer

<u>printerId</u>	p_type	location
------------------	--------	----------

Locker

<u>lockerId</u>	availability
-----------------	--------------

material_Borrow

<u>ID</u>	<u>Person_id</u>	<u>Mat_id</u>	date
-----------	------------------	---------------	------

material_Return

<u>ID</u>	<u>Person_id</u>	<u>Mat_id</u>	date
-----------	------------------	---------------	------

Settlements

<u>settlement_id</u>	<u>Mat_id</u>	<u>Person_id</u>	amount
----------------------	---------------	------------------	--------

order

<u>order_id</u>	order_name	order_amount	date
-----------------	------------	--------------	------

payment

<u>payment_id</u>	payment_amount	payment_method	date
-------------------	----------------	----------------	------

product

<u>product_id</u>	product_name	price
-------------------	--------------	-------

QuickShop_list

<u>order_id</u>	<u>product_id</u>	product_quantity	amount
-----------------	-------------------	------------------	--------

locker_Reserves

<u>locker_id</u>	<u>Person_id</u>	start_date	end_date
------------------	------------------	------------	----------

Phase 4:

Justification for 3NF

Employee:

<u>emp_id</u>	name	department	Position
---------------	------	------------	----------

In employee table, emp_id is the primary key and the other attributes are details of the employee which can be identified by the primary key that is emp_id. It does not depend on any attributes other than the primary key.

Staff:

<u>staff_id</u>	name	date_of_birth	department	Position
-----------------	------	---------------	------------	----------

In the staff table, staff_id is the primary key, and the other attributes are details of the staff that can be identified by the primary key which is staff_id. It does not depend on any attribute other than the primary key.

Student:

<u>stu_id</u>	name	date_of_birth	Dept	major
---------------	------	---------------	------	-------

In the student table, stu_id is the primary key, and the other attributes are details of the student which can be identified by the primary key which is stu_id. It does not depend on any attribute other than the primary key.

Books:

<u>Book_id</u>	b_name	author	Category	published_year	edition	Publication
----------------	--------	--------	----------	----------------	---------	-------------

In the books table, Book_id is the primary key, and the other attributes are details of the books which can be identified by the primary key which is Book_id . It does not depend on any attribute other than the primary key.

Journals/Articles/Magazines(J/A/M):

<u>J/A/M_id</u>	jam_name	Author	Category	published_year	Edition	publication
-----------------	----------	--------	----------	----------------	---------	-------------

In the books table, JAM_id is the primary key, and the other attributes are details of the JAM which can be identified by the primary key which is JAM_id. It does not depend on any attribute other than the primary key.

Master tables:

Person:

<u>Person_id</u>	<u>Person_Type</u>	<u>Emp_id</u>	<u>staff_id</u>	<u>stu_id</u>	name	Position
------------------	--------------------	---------------	-----------------	---------------	------	----------

The person table is a master table that is created such that the person_id is the Primary key and person_Type, Emp_id, staff_id, and stu_id are unique keys of the table. We have added name and Position attributes to the table attributes such that we can know the brief data about the person without looking it into the primary tables. Here the emp_id, staff_id, and stu_id are given as foreign key references such that there will not be any integrity issue when the triggers on the tables are removed.

Materials:

<u>Mat_id</u>	mat_type	<u>book_id</u>	<u>JAM_id</u>	name
---------------	----------	----------------	---------------	------

The material table is a master table that is created such that the mat_id is the Primary key and book_id, and JAM_id are unique keys of the table. We have added name attribute to the table attributes such that we can know the brief data about the material without looking it into the primary tables. Here the book_id and JAM_id are given as foreign key references such that there will not be any integrity issue when the triggers on the tables are removed.

Locker:

<u>lockerId</u>	availability
-----------------	--------------

In the locker table, lockerId is the primary key, and the other attributes are details of the locker that can be identified by the primary key which is locker_id. It does not depend on any attribute other than the primary key.

material_borrow:

<u>ID</u>	<u>Person_id</u>	<u>Mat_id</u>	date
-----------	------------------	---------------	------

In the material_Borrow table, Id is the primary key, and the other attributes are details of the material_Borrow that can be identified by the primary key which is id. It does not depend on any attribute other than the primary key.

material_Return:

<u>ID</u>	<u>Person_id</u>	<u>Mat_id</u>	date
-----------	------------------	---------------	------

In the material_Return table, Id is the primary key, and the other attributes are details of the material_Return that can be identified by the primary key which is Id. It does not depend on any attribute other than the primary key.

Settlements:

<u>settlement_id</u>	<u>Mat_id</u>	<u>Person_id</u>	amount
----------------------	---------------	------------------	--------

In the settlements table, settlement_id is the primary key, and mat_id and person_id are the unique values. In this table mat_id and person_id together with settlement_id form a candidate key such that we will get the information about what kind of settlements need to be done.

order:

<u>order_id</u>	order_name	order_amount	date
-----------------	------------	--------------	------

In the order table, order_id is the primary key, and the other attributes are details of the order that can be identified by the primary key which is order_id. It does not depend on any attribute other than the primary key.

payment:

<u>payment_id</u>	payment_amount	payment_method	date
-------------------	----------------	----------------	------

In the payment table, payment_id is the primary key, and the other attributes are details of the payment that can be identified by the primary key which is payment_id. It does not depend on any attribute other than the primary key.

product:

<u>product_id</u>	product_name	price
-------------------	--------------	-------

In the product table, product_id is the primary key, and the other attributes are details of the material_Return that can be identified by the primary key which is Id. It does not depend on any attribute other than the primary key.

QuickShop_list:

<u>order_id</u>	<u>product_id</u>	product_quantity	amount
-----------------	-------------------	------------------	--------

In the QuickShop_list table, order_id, and product_id together form the primary key, and the other attributes are details of the QuickShop_list that can be identified by the primary key. It does not depend on any attribute other than the primary key.

locker_Reserves:

<u>locker_id</u>	<u>Person_id</u>	start_date	end_date
------------------	------------------	------------	----------

In the locker_Reserves table,locker_id, and person_id together form the primary key, and the other attributes are details of the locker_Reserves that can be identified by the primary key which is Id. It does not depend on any attribute other than the primary key.

Data Dictionary

Person master table:

Table	Column	Data Type	Reference	Not null
Person	<u>person_id</u>	varchar(10)	Primary key	Y
Person	name	char(20)		Y
Person	DOB	Date		Y
Person	department	varchar(10)		Y
Person	Position	varchar(20)		Y
Person	Major	varchar(10)		N

Employee

Table	Column	Data Type	Reference	Not null
-------	--------	-----------	-----------	----------

employee	<u>emp_id</u>	varchar(10)	person_id(person)	Y
employee	name	char(20)		Y
employee	department	varchar(10)		Y
employee	Position	varchar(20)		Y

Staff

Table	Column	Data Type	Reference	Not null
staff	staff_id	varchar(10)	person_id(person)	Y
staff	name	char(20)		Y
staff	date_of_birth	Date		Y
staff	department	varchar(10)		Y
staff	Position	varchar(20)		Y

Student

Table	Column	Data Type	Reference	Not null
student	stu_id	varchar(10)	person_id(person)	Y
student	name	char(20)		Y
student	date_of_birth	Date		Y
student	Dept	varchar(10)		Y
student	major	varchar(10)		Y

Material

Table	Column	Data Type	Reference	Not null
Material	<u>mat_id</u>	varchar(10)	Primary Key	Y
Material	name	varchar(30)		Y
Material	author	char(20)		Y
Material	Category	char(20)		Y
Material	published_year	Date		Y

Material	edition	varchar(5)		N
Material	Publication	varchar(30)		Y

Books

Table	Column	Data Type	Reference	Not null
Books	Book_id	varchar(10)	mat_id (Material)	Y
Books	b_name	varchar(30)		Y
Books	author	char(20)		Y
Books	Category	char(20)		Y
Books	published_year	Date		Y
Books	edition	varchar(5)		N
Books	Publication	varchar(30)		Y

JAM

Table	Column	Data Type	Reference	Not null
J/A/M	Book_id	varchar(10)	mat_id (Material)	Y
J/A/M	b_name	varchar(30)		Y
J/A/M	author	char(20)		Y
J/A/M	Category	char(20)		Y
J/A/M	published_year	Date		Y
J/A/M	edition	varchar(5)		N
J/A/M	Publication	varchar(30)		Y

Table	Column	Data Type	Reference	Not null
Studyroom	roomId	int		Y
Studyroom	occupied	bit		Y
Studyroom	capacity	int		Y

Table	Column	Data Type	Reference	Not null
Studyroom_re	stuld	int	student	Y
Studyroom_re	roomId	int	studyroom	Y
Studyroom_re	Date	Date		Y
Studyroom_re	startTime	Time		Y
Studyroom_re	endTime	Time		Y

Table	Column	Data Type	Reference	Not null
Faculty	id	int		Y
Faculty	name	varchar(30)		Y
Faculty	department	varchar(30)		Y

Table	Column	Data Type	Reference	Not null
Lecturehall	hallId	int		Y
Lecturehall	occupied	bit		Y
Lecturehall	capacity	int		Y

Table	Column	Data Type	Reference	Not null
Lecturehall_re	facultyId	int	faculty	Y
Lecturehall_re	hallId	int	lecturehall	Y
Lecturehall_re	Date	Date		Y
Lecturehall_re	startTime	Time		Y
Lecturehall_re	endTime	Time		Y

Table	Column	Data Type	Reference	Not null
Lab	labId	int		Y
Lab	l_type	varchar(15)		Y
Lab	capacity	int		Y

Table	Column	Data Type	Reference	Not null
Printer	printerId	int		Y
Printer	location	varchar(30)		Y
Printer	p_type	varchar(15)		Y

Table	Column	Data Type	Reference	Not null
Guest	memberId	int		Y
Guest	name	varchar(30)		Y
Guest	phone	varchar(16)		Y

Table	Column	Data Type	Reference	Not null
membership	Id	int		Y
membership	member_id	int	guest	Y
membership	startDate	Date		Y
membership	endDate	Date		Y

Table	Column	Data Type	Reference	Not null
locker	locketr_id	int		Y

locker	availability	boolean		Y
--------	--------------	---------	--	---

Table	Column	Data Type	Reference	Not null
material_Borrow	id	int		Y
material_Borrow	Person_id	varchar[10]	Person	Y
material_Borrow	Mat_id	varchar[20]	Material	Y
material_Borrow	date	Date		Y

Table	Column	Data Type	Reference	Not null
material_Return	id	int		Y
material_Return	Person_id	varchar[10]	Person	Y
material_Return	Material_id	varchar[10]	Material	Y
material_Return	date	Date		Y

Table	Column	Data Type	Reference	Not null
settlement	settlement_id	int		Y
settlement	Person_id	varchar[10]	Person	Y

settlement	Material_id	varchar[10]	Material	Y
settlement	amount	int		Y

Table	Column	Data Type	Reference	Not null
order	order_id	varchar[10]		Y
order	order_name	varchar[20]		Y
order	order_amount	int		Y
order	date	Date		Y

Table	Column	Data Type	Reference	Not null
payment	payment_id	varchar[10]		Y
payment	payment_amount	int		Y
payment	payment_method	varchar[10]		Y
payment	date	Date		Y

Table	Column	Data Type	Reference	Not null
product	product_id	varchar[10]		Y
product	product_name	char[10]		Y
product	total_amount	int		Y

Table	Column	Data Type	Reference	Not null
QuickShop_list	order_id	varchar[10]	order	Y
QuickShop_list	product_id	varchar[10]	product	Y
QuickShop_list	product_quantity	int		Y
QuickShop_list	amount	int		Y

Table	Column	Data Type	Reference	Not null
locker_reserves	locker_id	int	locker	Y
locker_reserves	Person_id	varchar[10]	Person	Y
locker_reserves	start_date	Date		Y
locker_reserves	start_date	Date		Y

Unique keys

Type	Table	Key	Columns
Primary	Person	pk_Person	person_id
Primary	Matreial	pk_material	mat_id
Primary	studyroom	pk_studyroom	roomId
Primary	lecturehall	pk_lecuturehall	hallId
Primary	lab	pk_lab	labId
Primary	printer	pk_printer	printerId
Primary	Guest	pk_guest	memberId
Primary	membership	pk_membership	Id
Primary	locker	pk_locker	locker_id
Primary	borrow	pk_borrow	borrow_id
Primary	return	pk_return	return_id
Primary	settlement	pk_settlement	settlement_id
Primary	order	pk_printer	printer_id
Primary	payment	pk_payment	member_id
Primary	product	pk_product	Member_id

Foreign keys

Key	Table	Columns	ref-table	ref-column
fk_employee	employee	emp_id	Person	person_id
fk_staff	staff	staff_id	Person	person_id
fk_student	student	stu_id	Person	person_id
fk_books	books	book_id	Material	mat_id
fk_JAM	JAM	jam_id	Material	mat_id
fk_studyroom_re	studyroom_re	roomId	studyroom	roomId
fk_studyroom_re	studyroom_re	stuld	student	stuld
fk_lecturehall_re	lecturehall_re	hallId	lecturehall	hallId
fk_lecturehall_re	lecturehall_re	facultyId	faculty	facultyId
fk_membership	membership	memberId	guest	memberId
fk_locker_reserves	locker_reserves	Person_id	Person	Person_id
fk_locker_reserves	locker_reserves	locker_id	locker	locker_id
fk_borrow	borrow	Person_id	Person	Person_id
fk_borrow	borrow	Material_id	Material	Mat_id
fk_return	return	Person_id	Person	Person_id
fk_return	return	Material_id	Material	Mat_id
fk_settlement	settlement	Person_id	Person	Person_id
fk_settlement	settlement	Material_id	Material	Mat_id
fk_QuickShop_list	QuickShop_list	order_id	order	order_id
fk_QuickShop_list	QuickShop_list	product_id	product	product_id

summary

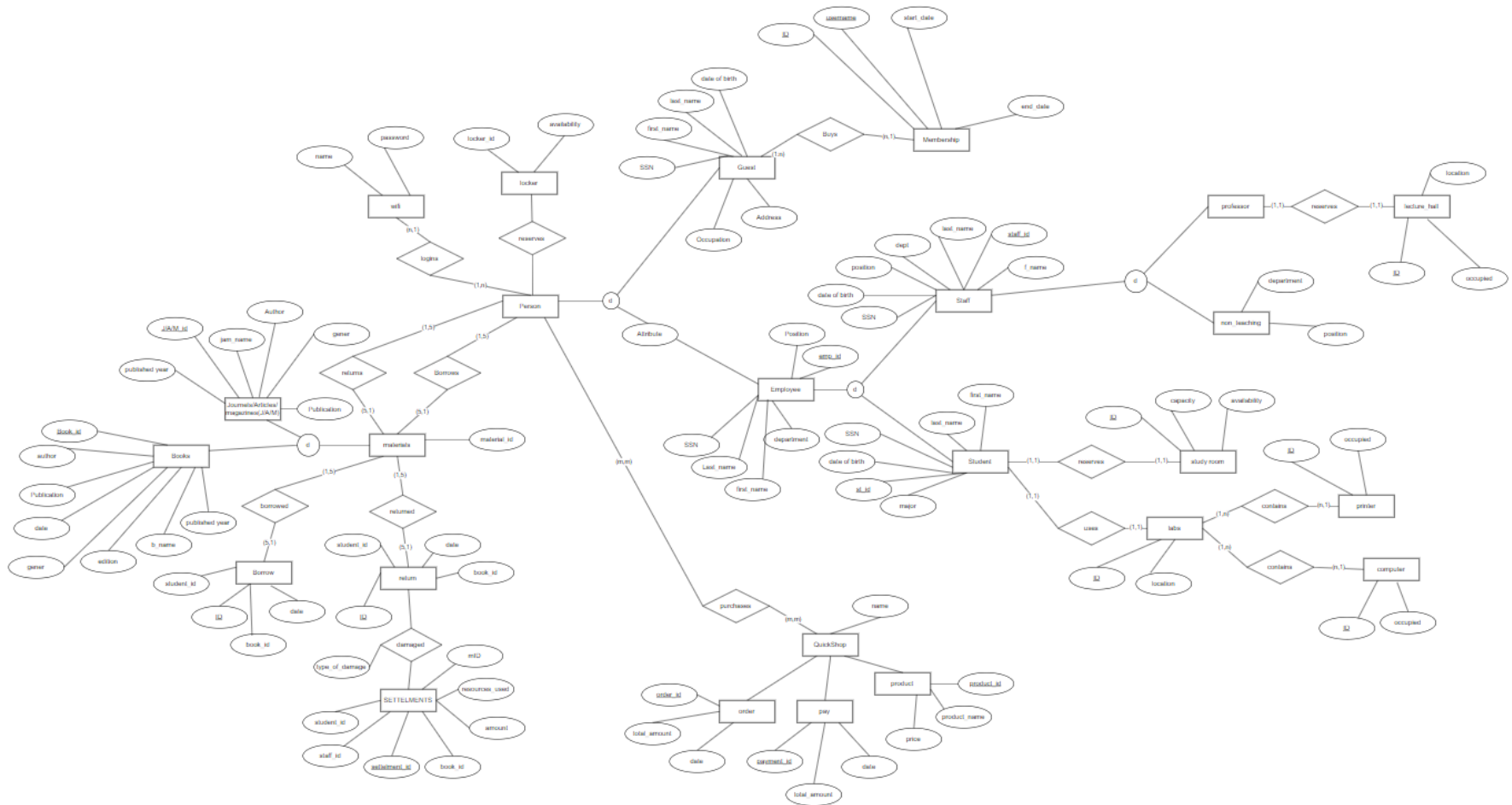
For better management of library resources, record students' borrowing and returning books, and better allocation of resources like library study rooms, we create an easy access and management database system. This database can be used to record the students' details, borrow and return books, the reservation of study rooms and lecture halls, and so on. We have 18 entities in total.

Teamwork

We separated the work into three parts, and each of us designed several entities (including the attributes and the relationships, and the partial ER model), and wrote the requirements of the entities. Then we combined our work together, the TA might have noticed that there are some different formats in the relationships part. It is because we draw the figure separately.

s.No	Work	Done By	Description
1	Introduction, design Requirements, functional requirements.	Preethi, Yagnashree, Peiyu Li	All three of us discussed the details and came to a conclusion.
2	entities	Yagnashree works on entities 1-7 Peiyu Li works on entities 8-14 Preethi works on entities 15-20	Each one of us worked on several entities
3	ER diagram	Yagnashree, Peiyu Li, Preethi	
4	3NF	Yagnashree	Entities 1-7
5	3NF	Peiyu Li	Entities 8-14
6	3NF	Preethi	Entities 15-23

Appendix A - ER diagram



The picture we show here is not clear enough to identify, so please find the image of ER diagram here and zoom in to get a clear image:
[ER_Diagram.jpg](#)

Relational model:

