LIBRARY MANAGEMENT SYSTEM

Abstract:

Library management refers to the issues involved in managing the resources available in library such as classification of books, books processing and borrowing books. There are different systems employed by libraries to manage resources in an efficient manner but still there are many problems so many of them switch to new systems.

In our system we have books, members and staff entities with there respective attributes. The borrowing details of a book and to store the details of pending borrowed books we have borrow and pending table. To store the who;e borrow and return status of a book we have a history table.

Requirement Analysis:

Existing System:

Here we have our GITAM KRC library management system

Difficulties with manual library administration.

- Search for a particular book takes a lot of time.
- Keeping track of how many quantities of each book is available.
- Unable to keep track of old books that need to be replaced.
- Keeping track of renewals is difficult.
- Unable to keep track of history of a book.
- Keeping track of lost books
- Identifying who has borrowed a book is difficult.
- Keeping track of books requested by students.
- Alerting student to renew his/her books.
- Staff details.

Proposed System:

Every book will be assigned a unique ID so that each and every book can be tracked with the LMS. So with the help of the ID we can immediately see the current borrower and as well as how many books of similar kind are in stock along with the past history of the borrowers. Based on the date of the book purchased by the library, the software will remind the librarians about the age of the book and if it has to be replaced.

The librarian can also store the current status of the book based on 4 categories, namely, NEW,

MARKED, READABLE, OLD.

The LMS will have all the details of the student and the librarian can also rate the students on

how they maintain the borrowed books and based on the student rating they can decide

whether they are eligible to borrow very valuable and rare book copies that are not easily

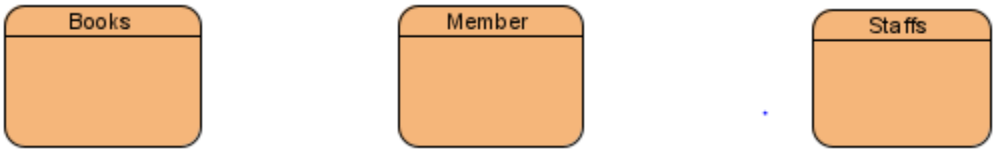
available.

ER Model:

Step 1. Entity Identification:

We have entities called

- Staff
- Books
- Members
- Pending returns



Step 2. Relationship Identification:

We have the following two relationships

- The Staff manages Books **borrowed** by Members.
- The Members **borrows** Books.
- Pending of Books record in **pending.**

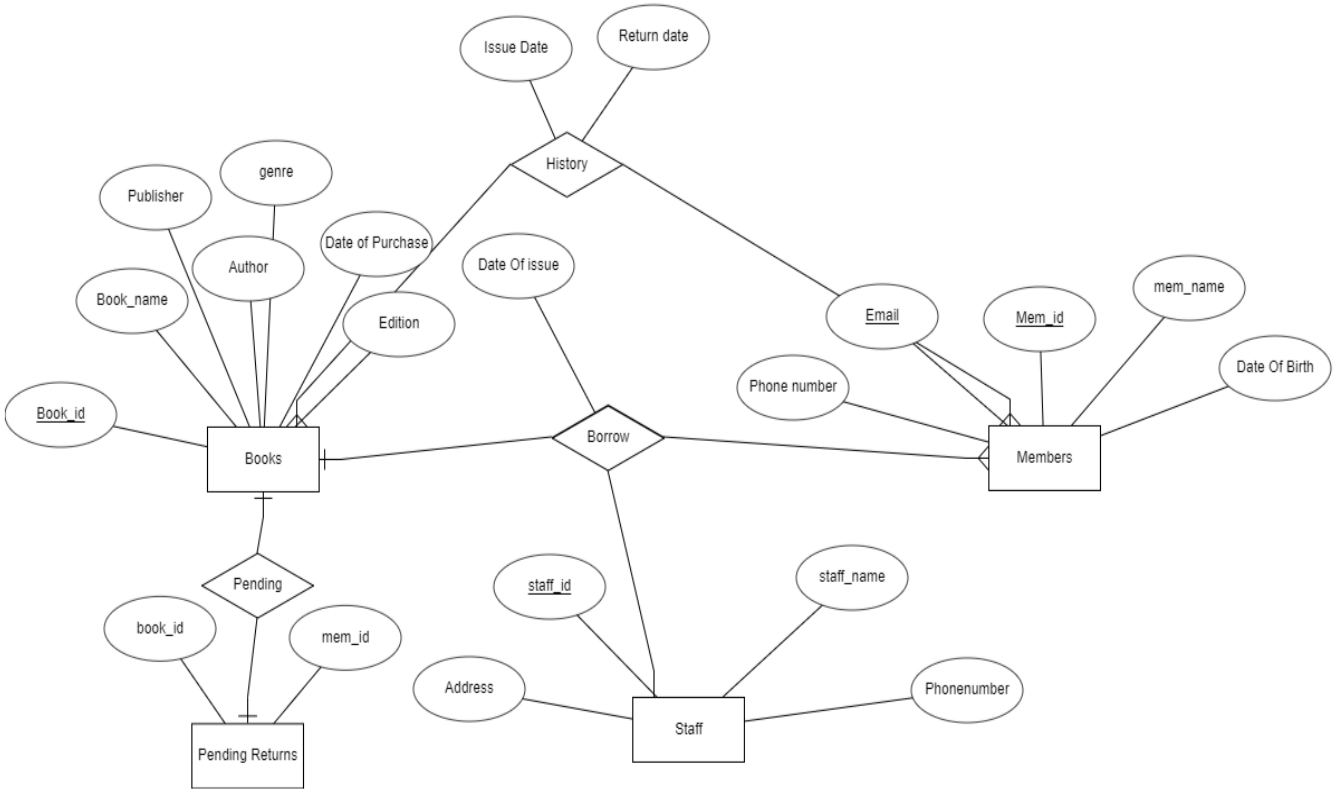
Step 3. Cardinality Identification:

- The Staff manages multiple borrows of Books.
- The Members borrows multiple Books.
- There are multiple Books in pending.

Step 4. Identify Attributes

Entity	***Primary key***	***Attributes***
Books	Book ID(bid)	Book name, author, publisher, genre, edition, date of purchase
Staff	Staff ID(sid)	Staff name, address, phone number
Members	Mem ID(mid)	Mem name, email, phone number, DOB

Step 5. Create the ERD:



NORMALISATION:

Normalization is the process of minimizing **redundancy** from a relation or set of relations, Redundancy in relation may cause insertion, deletion and update anomalies.

We have Functional Dependencies (FD) for Library Management they are

Bid → Bname, author, publisher, genre, edition, dateofpurcahse

Sid → Sname, address, phone number

Mid → Mname, email, phone number, DateOfbirth

Bid → DOI

First Normal Form (1NF):

A relation is in first normal form if every attribute in that relation is singled valued attribute.

Here in our above FDs in which are not encouraging composite or multi valued attributes. Therefore our scenario is in 1NF.

Second Normal Form (2NF):

A relation to be in second normal form,

- the relation must be in first normal form and
- Relation must not contain any partial dependency (no non-prime attribute is dependent on any candidate key).

Here our FDs are in 1NF.

Now we check for Essential attributes. From our clear observation we consider Bid, Mid, Sid as Essential key.

From closure of (Bid, Mid, Sid) we get all attributes.so our essentials attributes are combination of Bid, Mid, Sid.

Let consider,

Prime Attributes: Mid, Sid, Bid

Nonprime attributes: Bname, author, publisher, genre, edition, dateofpurcahse, Sname, address, phone number, Mname, email, phone number, DateOfbirth, DOI.

In first FD: **Bid → Bname, author, publisher, genre, edition, dateofpurcahse**

Bid is part of prime and *Bname, author, publisher, genre, edition, dateofpurcahse* are part of nonprime attributes.

Therefore it has **partial Dependence.**

In second FD: **Sid → Sname, address, phone number**

Sid is part of prime and *Sname, address, phone number* are part of nonprime attributes.

Therefore it has **partial Dependence.**

In third FD: **Mid → Mname, email, phone number, DateOfbirth**

Mid is part of prime and *Mname, email, phone number, DateOfbirth* are part of nonprime attributes.

Therefore it has **partial Dependence.**

In fourth FD: **Bid → DOI**

Bid is part of prime and *DOI* is part of nonprime attributes.

Therefore it has **partial Dependence.**

Our Relation to be in 2NF we need to create separate tables for all the four FDs we considered.

By this we can conclude our scenario is in 2NF.

Third Normal Form (3NF):

A relation to be in Third normal form,

- the relation must be in second normal form and
- if there is no transitive dependency

If X → Y and Y → Z are two FDs then X → Z is called transitive dependency.

Here our given FDs are in 2NF.

From the above definition of transitive dependency we can clearly observe that there is no such transitive dependencies in our above mentioned FDs.

Therefore our relations are in 3NF.

Boyce codd Normal Form (BCNF):

BCNF is the advance version of 3NF.

A relation to be in Boyce codd normal form,

- The relation should be in 3NF, and
- if every functional dependency X → Y, X is the super key of the table.

Here our FDs are in 3NF.

From our FDs provided we can say that all attributes in the LHS side are super keys of the respective table.

Therefore our relations are in BCNF.

At last we concluded that we have these tables.

- **BOOKS** table with attributes **Bid, Bname, author, publisher, genre, edition, dateofpurcahse** and ***BID*** as primary key.
- **MEMBERS** table with attributes **Mid, Mname, email, phone number, DateOfbirth** and ***Mid*** as primary key.
- **STAFF** table with attributes **Sid, Sname, address, phone number** and ***Sid*** as primary key.
- **BORROWS** table with attributes Bid, Mid, Sid, DOI as ***BID*** primary key.

SQL QUERIES:

```
create table Books(bid integer primary key,bname varchar(30),author varchar(30),edi varchar(30),pub varchar(30),genre varchar(30),dop date);

create table Members(mid integer primary key,mname varchar(30),email varchar(30),phno varchar(30),dob date);

create table staff(sid integer primary key,sname varchar(30),address varchar(40),phno varchar(30));

create table borrows(bid integer,mid integer,sid integer,doi date,foreign key (bid) references Books(bid),foreign key (mid) references Members(mid),foreign key (sid) references staff(sid),primary key(bid));

create table pendings(bid integer,mid integer,foreign key (bid) references Books(bid),foreign key (mid) references Members(mid),primary key(bid,mid));

create table history(bid integer,mid integer,doi date, dor char(40),foreign key (bid) references Books(bid),foreign key (mid) references Members(mid),primary key(bid,mid,doi,dor));

create or replace trigger trigger1 after insert on borrows for each row

declare bid1 integer;mid1 integer;

begin

bid1:=:new.bid;

mid1:=:new.mid;

insert into pendings values(bid1,mid1);

end;

insert into Books values(1001,'C How to program','Harvey Deitel','1','Prentice Hall','general','04-OCT-2000');

insert into Books values(1002,'C How to program','Harvey Deitel','2','Prentice Hall','general','09-SEP-2002');

insert into Books values(1003,'C How to program','Harvey Deitel','3','Prentice Hall','general','23-AUG-2003');
```

insert into Books values(1021,'JAVA How to program','maniesh agraval','3','Prentice Hall','general','29-MAY-1997');

insert into Books values(1031,'Internet of things','somesh yeshen','3','kingstroms publishers','general','09-DEC-1997');

insert into Books values(1032,'Internet of things','Ram shan','1','lamba publishers','general','09-DEC-2001');

insert into Books values(1041,'Database management system','Ramakrishanan','5','Jaico Publishing House','general','29-MAY-1997');

insert into Books values(1051,'Python programming','uday chandran','1','lionpolish homes','general','13-OCT-2000');

insert into Books values(1052,'Python programming','uday chandran','2','lionpolish homes','general','13-OCT-2001');

insert into Books values(1061,'Artifical inteligence','michel jan','4','standhouse','general','08-FEB-2007');

select * from books;

BID	BNAME	AUTHOR	EDI	PUB	GENRE	DOP
1001	C How to program	Harvey Deitel	1	Prentice Hall	general	04-OCT-00
1002	C How to program	Harvey Deitel	2	Prentice Hall	general	09-SEP-02
1003	C How to program	Harvey Deitel	3	Prentice Hall	general	23-AUG-03
1021	JAVA How to program	maniesh agraval	3	Prentice Hall	general	29-MAY-97
1031	Internet of things	somesh yeshen	3	kingstroms publishers	general	09-DEC-97
1032	Internet of things	Ram shan	1	lamba publishers	general	09-DEC-01
1041	Database management system	Ramakrishanan	5	Jaico Publishing House	general	29-MAY-97
1051	Python programming	uday chandran	1	lionpolish homes	general	13-OCT-00
1052	Python programming	uday chandran	2	lionpolish homes	general	13-OCT-01
1061	Artifical inteligence	michel jan	4	standhouse	general	08-FEB-07

insert into Members values(1214004,'Anirudh','anun004@gmail.com','923457','08-AUG-2005');

insert into Members values(1214005,'Daksh','dak005@gmail.com','893457','15-AUG-2006');

insert into Members values(1214018,'Jatin','iamjatin5@gmail.com','893452','12-OCT-2006');

insert into Members values(1214031,'Ketan','saiketan31@gmail.com','985624','02-JAN-2005');

insert into Members values(1214012,'Arjun','mynamearjun12@gmail.com','898923','16-MAY-2005');

insert into Members values(1214003,'Karan','karanadaran3@gmail.com','876852','26-SEP-2006');

insert into Members values(1214056,'Nayan','nayansrinivas56@gmail.com','865729','03-OCT-2005');

insert into Members values(1214047,'Gaurav','gauravsan47@gmail.com','945629','12-OCT-2005');

insert into Members values(1214058,'Tanay','tonysin58@gmail.com','976789','13-JAN-2005');

insert into Members values(1214011,'Zaher','khanzaher11@gmail.com','934562','13-DEC-2005');

select * from members;

MID	MNAME	EMAIL	PHNO	DOB
1214004	Anirudh	anun004@gmail.com	923457	08-AUG-05
1214005	Daksh	dak005@gmail.com	893457	15-AUG-06
1214018	Jatin	iamjatin5@gmail.com	893452	12-OCT-06
1214031	Ketan	saiketan31@gmail.com	985624	02-JAN-05
1214012	Arjun	mynamearjun12@gmail.com	898923	16-MAY-05
1214003	Karan	karanadaran3@gmail.com	876852	26-SEP-06
1214056	Nayan	nayansrinivas56@gmail.com	865729	03-OCT-05
1214047	Gaurav	gauravsan47@gmail.com	945629	12-OCT-05
1214058	Tanay	tonysin58@gmail.com	976789	13-JAN-05
1214011	Zaher	khanzaher11@gmail.com	934562	13-DEC-05

insert into staff values(2200001,'vamesh','vizag','986312');

insert into staff values(2200011,'darma','vizag','976549');

insert into staff values(2200021,'jashwanth kumar','vizag','768543');

insert into staff values(2200003,'latha','vizag','959884');

insert into staff values(2200025,'sanjana','vizag','630344');

select * from staff;

SID	SNAME	ADDRESS	PHNO
2200001	vamesh	vizag	986312
2200011	darma	vizag	976549
2200021	jashwanth kumar	vizag	768543
2200003	latha	vizag	959884
2200025	sanjana	vizag	630344

insert into borrows values(1001,1214004,2200001,'09-DEC-2020');

insert into borrows values(1021,1214004,2200001,'09-DEC-2020');

```
insert into borrows values(1031,1214018,2200011,'07-DEC-2020');
insert into borrows values(1041,1214018,2200025,'06-DEC-2020');
insert into borrows values(1002,1214003,2200021,'07-DEC-2020');
insert into borrows values(1061,1214056,2200011,'08-DEC-2020');
insert into borrows values(1032,1214056,2200003,'08-DEC-2020');
insert into borrows values(1051,1214004,2200003,'08-DEC-2020');
insert into borrows values(1052,1214012,2200025,'09-DEC-2020');
select * from borrows;
```

BID	MID	SID	DOI
1001	1214004	2200001	09-DEC-20
1021	1214004	2200001	09-DEC-20
1031	1214018	2200011	07-DEC-20
1041	1214018	2200025	06-DEC-20
1002	1214003	2200021	07-DEC-20
1061	1214056	2200011	08-DEC-20
1032	1214056	2200003	08-DEC-20
1051	1214004	2200003	08-DEC-20
1052	1214012	2200025	09-DEC-20

```
select * from pendings;
```

BID	MID
1001	1214004
1002	1214003
1021	1214004
1031	1214018
1032	1214056
1041	1214018
1051	1214004
1052	1214012
1061	1214056

```
create or replace trigger trigger2 after delete on borrows for each row
declare bid1 integer;mid1 integer;doi1 date;dor1 char(40);
begin
bid1:=:old.bid;
mid1:=:old.mid;
doi1:=:old.doi;
dor1:=to_char(sysdate,'DD-MON-YYYY HH:MI:SS AM');
insert into history values(bid1,mid1,doi1,dor1);
delete from pendings p1 where p1.bid = bid1;
end;
delete from borrows where bid = 1021;
select * from history;
```

select **distinct** bname from Books;

BNAME
C How to program
Python programming
Internet of things
Database management system
JAVA How to program
Artifical inteligence

select **count**(distinct bname) from Books;

COUNT(DISTINCTBNAME)
6

select bname,count(bname) from Books **group by** bname;

BNAME	COUNT(BNAME)
C How to program	3
Python programming	2
Internet of things	2
Database management system	1
JAVA How to program	1
Artifical inteligence	1

select bname,edi from Books **where** bname = 'C How to program';

BNAME	EDI
C How to program	1
C How to program	2
C How to program	3

select * from Members **order by** mid asc;

MID	MNAME	EMAIL	PHNO	DOB
1214003	Karan	karanadaran3@gmail.com	876852	26-SEP-06
1214004	Anirudh	anun004@gmail.com	923457	08-AUG-05
1214005	Daksh	dak005@gmail.com	893457	15-AUG-06
1214011	Zaher	khanzaher11@gmail.com	934562	13-DEC-05
1214012	Arjun	mynamearjun12@gmail.com	898923	16-MAY-05
1214018	Jatin	iamjatin5@gmail.com	893452	12-OCT-06
1214031	Ketan	saiketan31@gmail.com	985624	02-JAN-05
1214047	Gaurav	gauravsan47@gmail.com	945629	12-OCT-05
1214056	Nayan	nayansrinivas56@gmail.com	865729	03-OCT-05
1214058	Tanay	tonysin58@gmail.com	976789	13-JAN-05

create **view** borrowersview as select * from borrows where doi **between** '02-DEC-2020' and '08-DEC-2020';

select * from borrowersview;

BID	MID	SID	DOI
1031	1214018	2200011	07-DEC-20
1041	1214018	2200025	06-DEC-20
1002	1214003	2200021	07-DEC-20
1061	1214056	2200011	08-DEC-20
1032	1214056	2200003	08-DEC-20
1051	1214004	2200003	08-DEC-20

select bname,author from Books where bname='C How to program' **and** edi='3';

BNAME	AUTHOR
C How to program	Harvey Deitel

select bname,author from Books where bname='C How to program' **or** edi='3';

BNAME	AUTHOR
C How to program	Harvey Deitel
C How to program	Harvey Deitel
C How to program	Harvey Deitel
JAVA How to program	maniesh agraval
Internet of things	somesh yeshen

select bname,author,edi from Books where bname **like** 'C%';

BNAME	AUTHOR	EDI
C How to program	Harvey Deitel	1
C How to program	Harvey Deitel	2
C How to program	Harvey Deitel	3

select * from Books where **not** bname='C How to program';

BID	BNAME	AUTHOR	EDI	PUB	GENRE	DOP
1021	JAVA How to program	maniesh agraval	3	Prentice Hall	general	29-MAY-97
1031	Internet of things	somesh yeshen	3	kingstroms publishers	general	09-DEC-97
1032	Internet of things	Ram shan	1	lamba publishers	general	09-DEC-01
1041	Database management system	Ramakrishanan	5	Jaico Publishing House	general	29-MAY-97
1051	Python programming	uday chandran	1	lionpolish homes	general	13-OCT-00
1052	Python programming	uday chandran	2	lionpolish homes	general	13-OCT-01
1061	Artifical inteligence	michel jan	4	standhouse	generàl	08-FEB-07

select bname,**max**(edi) from Books group by bname;

BNAME	MAX(EDI)
C How to program	3
Python programming	2
Internet of things	3
Database management system	5
JAVA How to program	3
Artifical inteligence.	4

select * from Members m **inner join** borrows b1 on m.mid = b1.mid;

MID	MNAME	EMAIL	PHNO	DOB	BID	MID	SID	DOI
1214004	Anirudh	anun004@gmail.com	923457	08-AUG-05	1001	1214004	2200001	09-DEC-20
1214004	Anirudh	anun004@gmail.com	923457	08-AUG-05	1021	1214004	2200001	09-DEC-20
1214004	Anirudh	anun004@gmail.com	923457	08-AUG-05	1051	1214004	2200003	08-DEC-20
1214018	Jatin	iamjatin5@gmail.com	893452	12-OCT-06	1031	1214018	2200011	07-DEC-20
1214018	Jatin	iamjatin5@gmail.com	893452	12-OCT-06	1041	1214018	2200025	06-DEC-20
1214012	Arjun	mynamearjun12@gmail.com	898923	16-MAY-05	1052	1214012	2200025	09-DEC-20
1214003	Karan	karanadaran3@gmail.com	876852	26-SEP-06	1002	1214003	2200021	07-DEC-20
1214056	Nayan	nayansrinivas56@gmail.com	865729	03-OCT-05	1061	1214056	2200011	08-DEC-20
1214056	Nayan	nayansrinivas56@gmail.com	865729	03-OCT-05	1032	1214056	2200003	08-DEC-20

select * from Members m **left join** borrows b1 on m.mid = b1.mid;

MID	MNAME	EMAIL	PHNO	DOB	BID	MID	SID	DOI
1214004	Anirudh	anun004@gmail.com	923457	08-AUG-05	1001	1214004	2200001	09-DEC-20
1214004	Anirudh	anun004@gmail.com	923457	08-AUG-05	1021	1214004	2200001	09-DEC-20
1214018	Jatin	iamjatin5@gmail.com	893452	12-OCT-06	1031	1214018	2200011	07-DEC-20
1214018	Jatin	iamjatin5@gmail.com	893452	12-OCT-06	1041	1214018	2200025	06-DEC-20
1214003	Karan	karanadaran3@gmail.com	876852	26-SEP-06	1002	1214003	2200021	07-DEC-20
1214056	Nayan	nayansrinivas56@gmail.com	865729	03-OCT-05	1061	1214056	2200011	08-DEC-20
1214056	Nayan	nayansrinivas56@gmail.com	865729	03-OCT-05	1032	1214056	2200003	08-DEC-20
1214004	Anirudh	anun004@gmail.com	923457	08-AUG-05	1051	1214004	2200003	08-DEC-20
1214012	Arjun	mynamearjun12@gmail.com	898923	16-MAY-05	1052	1214012	2200025	09-DEC-20
1214011	Zaher	khanzaher11@gmail.com	934562	13-DEC-05	-	-	-	-
1214058	Tanay	tonysin58@gmail.com	976789	13-JAN-05	-	-	-	-
1214031	Ketan	saiketan31@gmail.com	985624	02-JAN-05	-	-	-	-
1214005	Daksh	dak005@gmail.com	893457	15-AUG-06	-	-	-	-
1214047	Gaurav	gauravsan47@gmail.com	945629	12-OCT-05	-	-	-	-

select * from borrows b1 **right joisn** Members m on m.mid = b1.mid;

BID	MID	SID	DOI	MID	MNAME	EMAIL	PHNO	DOB
1001	1214004	2200001	09-DEC-20	1214004	Anirudh	anun004@gmail.com	923457	08-AUG-05
1021	1214004	2200001	09-DEC-20	1214004	Anirudh	anun004@gmail.com	923457	08-AUG-05
1031	1214018	2200011	07-DEC-20	1214018	Jatin	iamjatin5@gmail.com	893452	12-OCT-06
1041	1214018	2200025	06-DEC-20	1214018	Jatin	iamjatin5@gmail.com	893452	12-OCT-06
1002	1214003	2200021	07-DEC-20	1214003	Karan	karanadaran3@gmail.com	876852	26-SEP-06
1061	1214056	2200011	08-DEC-20	1214056	Nayan	nayansrinivas56@gmail.com	865729	03-OCT-05
1032	1214056	2200003	08-DEC-20	1214056	Nayan	nayansrinivas56@gmail.com	865729	03-OCT-05
1051	1214004	2200003	08-DEC-20	1214004	Anirudh	anun004@gmail.com	923457	08-AUG-05
1052	1214012	2200025	09-DEC-20	1214012	Arjun	mynamearjun12@gmail.com	898923	16-MAY-05
-	-	-	-	1214011	Zaher	khanzaher11@gmail.com	934562	13-DEC-05
-	-	-	-	1214058	Tanay	tonysin58@gmail.com	976789	13-JAN-05
-	-	-	-	1214031	Ketan	saiketan31@gmail.com	985624	02-JAN-05
-	-	-	-	1214005	Daksh	dak005@gmail.com	893457	15-AUG-06
-	-	-	-	1214047	Gaurav	gauravsan47@gmail.com	945629	12-OCT-05

select b.bname from Books b where b.bid **in**(select b1.bid from borrows b1 where b1.mid = 1214004);

BNAME
C How to program
JAVA How to program
Python programming

select b.bname from Books b where b.bid **in**(select b1.bid from borrows b1 where b1.mid **=**(select m.mid from Members m where m.mname ='Jatin'));

BNAME
Internet of things
Database management system

select m.mname from Members m where m.mid **not in**(select b1.mid from borrows b1);

MNAME
Zaher
Tanay
Ketan
Daksh
Gaurav

select b.bname from Books b where exists(select * from borrows b1 where b1.mid = 1214004 and b.bid = b1.bid);

BNAME
C How to program
JAVA How to program
Python programming

delete from borrows where bid = 1021;

select * from history;

BID	MID	DOI	DOR
1021	1214004	09-DEC-20	12-DEC-2020 03:42:30 PM