AMERICAN INTERNATIONAL UNIVERSITY BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

Course Title

INTRODUCTION TO DATABASE

FALL 2023-2024
Section: L

TITLE

"LIBRARY MANAGEMENT SYSTEM"

Supervised By

MD Sajid Bin Faisal

Submitted By: Group no: 6

Name	ID
IRTISAM FARUQUI ALAVI	22-48863-3
MD.FAZLEH RABBI YAHIA PRANTO	22-49575-3
AMIT DATTA DIP	22-48860-3
RIASAD CHOUDHURY	22-49093-3

TABLE OF CONTENTS

1. INTRODUCTION :-

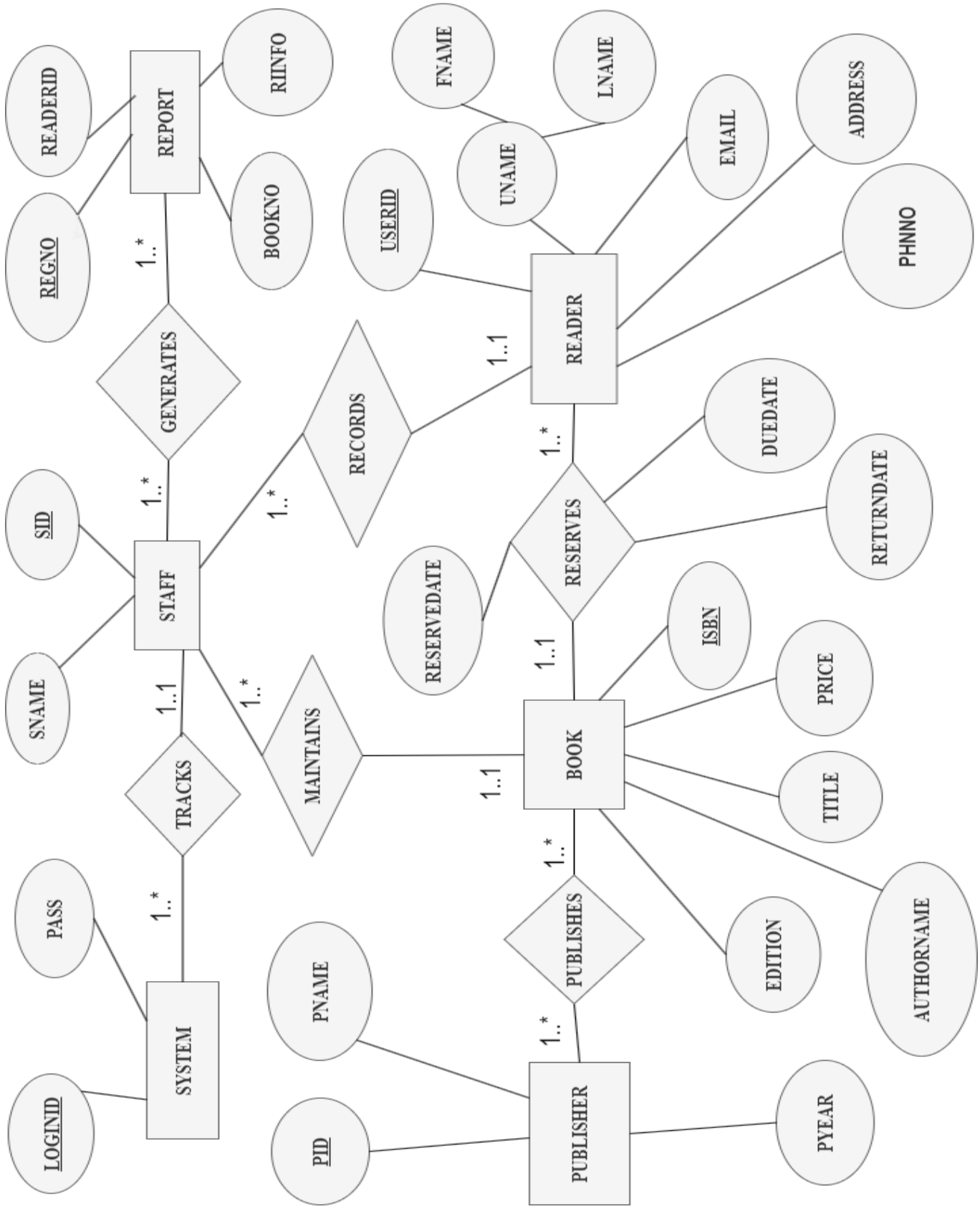
This database project which is being implemented by making a schema namely "librarymgt" firstly, focuses on the library management system. In the era of technological revolution, this database system is essentially required in the field of education, initiating smoothness in library management. The report is thereby made with the aid of draw.io,oracle database platform and many other tools accordingly.

2. CASE STUDY :-

LIBRARY MANAGGENT SYSTEM

The library management system keeps track of the staff with authentication system compressing login ID and password. The system provides login to multiple staff. Staff has its own unique identity number and name. Staff maintains the book catalog with its ISBN, Book title, price, category, Author number and details and edition. A staff maintains multiple books. A publisher can publish many books, but a book is published by only one publisher. A publisher has publisher ID, year when the book was published and his name. The staff keeps record of the readers. The readers are registered with their user ID, email, name, including fast name, last name, phone numbers and address. Readers can reserve books that stamps with reserve date and return date. If not returned within due time, it can have a due date also. A reader can reserve many books, but one book can be reserved by only one reader. Staff also generate multiple reports that has reader id, registration no of report, book no and return issue info. A staff can generate many reports and many reports can be managed by one or more staff.

3. ER DIAGRAM :-


READERID
REGNO
REPORT
RIINFO
BOOKNO
1..*
GENERATES
1..*
SID
STAFF
SNAME
1..1
TRACKS
1..*
PASS
SYSTEM
LOGINID
1..*
RECORDS
1..*
MAINTAINS
1..1
USERID
READER
1..1
FNAME
UNAME
LNAME
EMAIL
ADDRESS
PHNNO
1..*
RESERVES
RESERVEDATE
DUEDATE
RETURNDATE
1..1
BOOK
ISBN
PRICE
TITLE
EDITION
AUTHORNAME
1..*
PUBLISHES
1..*
PNAME
PID
PUBLISHER
PYEAR

4. NORMALIZATION :-

- RECORDS:

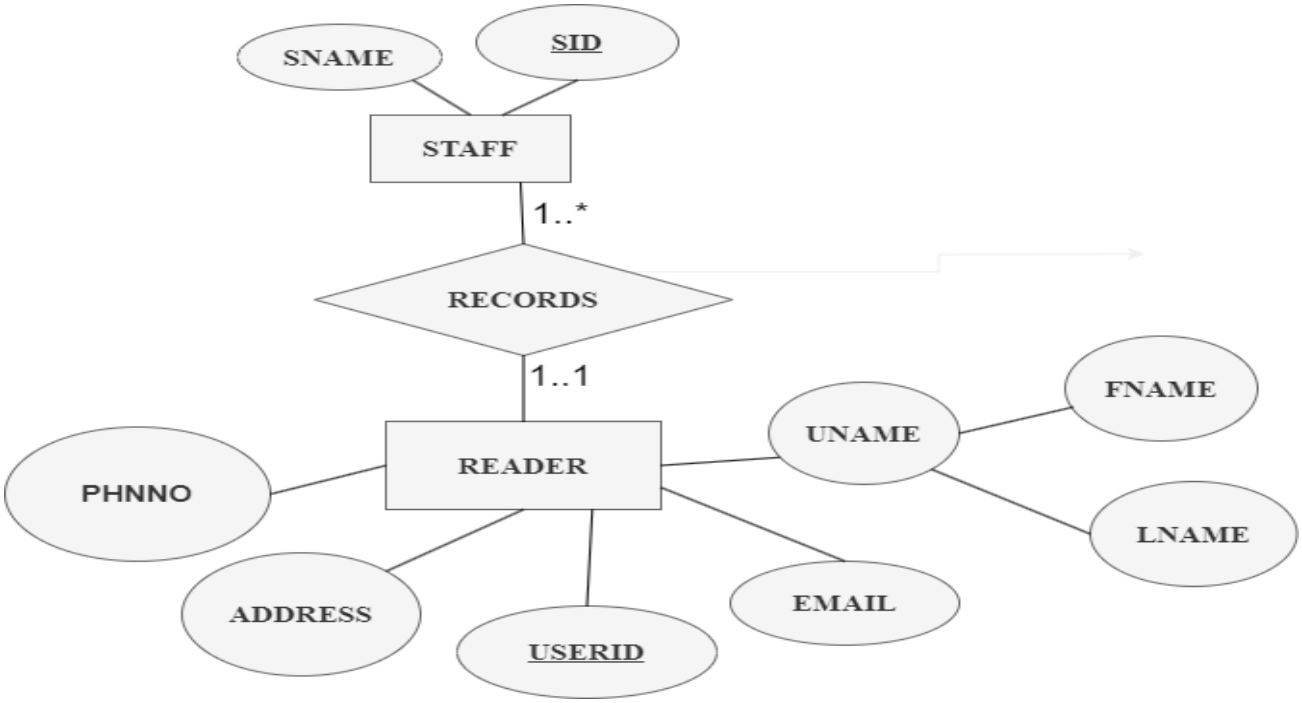


Fig 4.1 Records relation

UNF: SName, SID, UserID, Email, PhnNo, Address, UName, FName, LName

1NF: SName, SID, UserID, Email, PhnNo, Address, UName, FName, LName

2NF: 1. SName, SID(PK), UserID(FK)

2. UserID(PK), Email, PhnNo, Address, FName, LName

3NF: As Same as 2NF

- MAINTAINS:

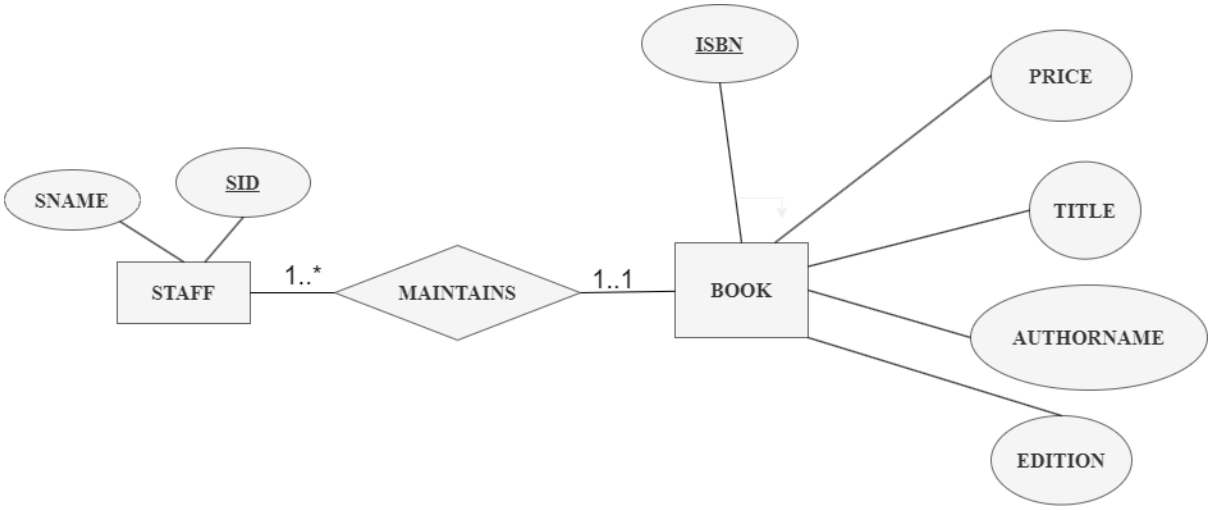


Fig 4.2 Maintains relation

UNF: SName, SID, Edition, AuthorName, Title, Price, ISBN

1NF: SName, SID, Edition, AuthorName, Title, Price, ISBN

2NF: 1. SName, SID(PK), ISBN(FK)

2. ISBN(PK), Edition, AuthorName, Title, Price

3NF: As Same as 2NF

- TRACKS

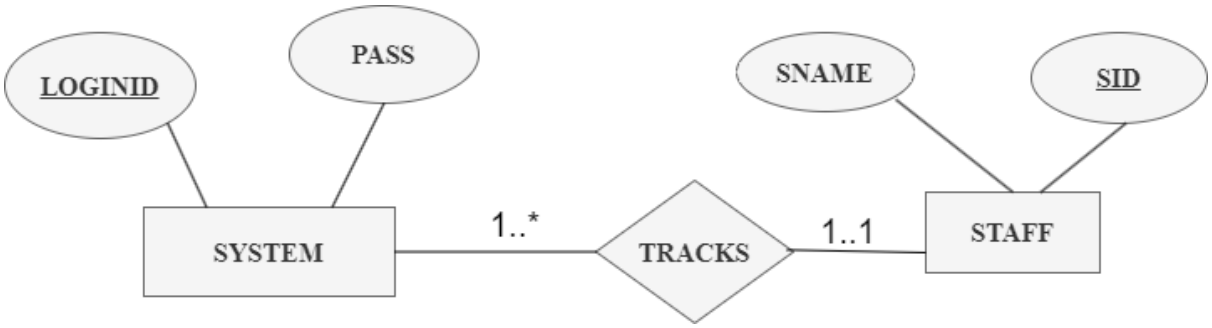


Fig 4.3 Tracks relation

UNF: LoginID, Pass, SName, SID

1NF: LoginID, Pass, SName, SID

2NF: 1. SID(PK), SName

2. LoginID(PK), Pass, SID(FK)

3NF: As Same as 2NF

- PUBLISHES:

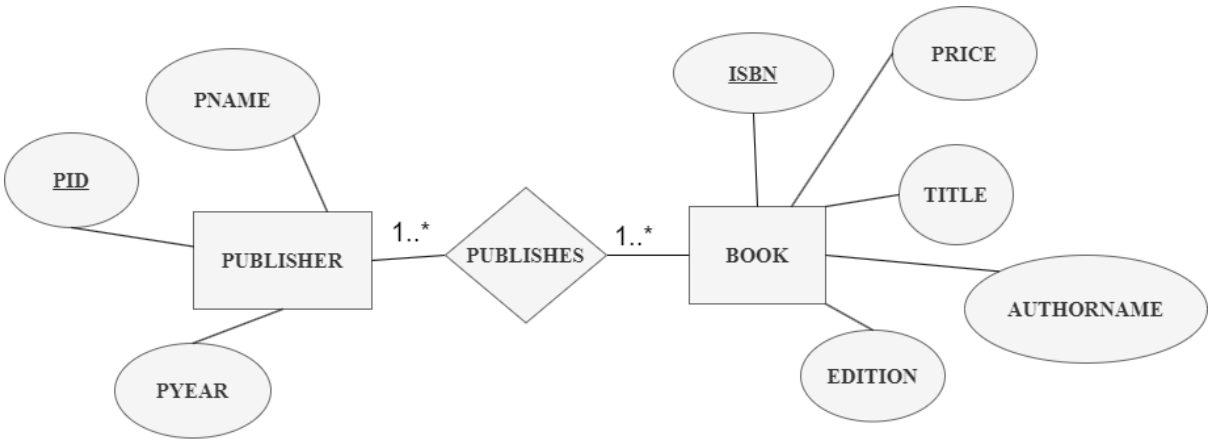


Fig 4.4 Publishes relation

UNF: PID, PName, PYear, Edition, AuthorName, Title, ISBN,Price

1NF: PID, PName, PYear, Edition, AuthorName, Title, ISBN,Price

2NF: 1. PID(PK), PName, PYear

2. ISBN(PK), Price, Title, Edition, AuthorName

3. PID(PK), ISBN(FK)

3NF: As Same as 2NF

- GENERATES:

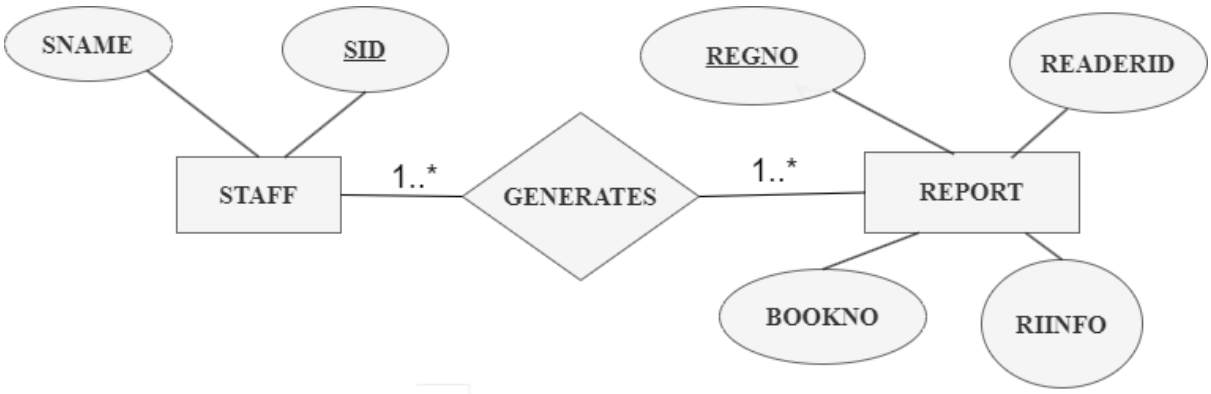


Fig 4.5 Generates relation

UNF: SName,SID,RegNo,ReaderID,Bookno,RIinfo

1NF: SName,SID,RegNo,ReaderID,Bookno,RIinfo

2NF: 1. SID(PK),SName

2. RegNo(PK),ReaderID,Bookno,RIinfo

3. SID(PK),RegNo(FK)

3NF: As same as 2NF

- RESERVES:

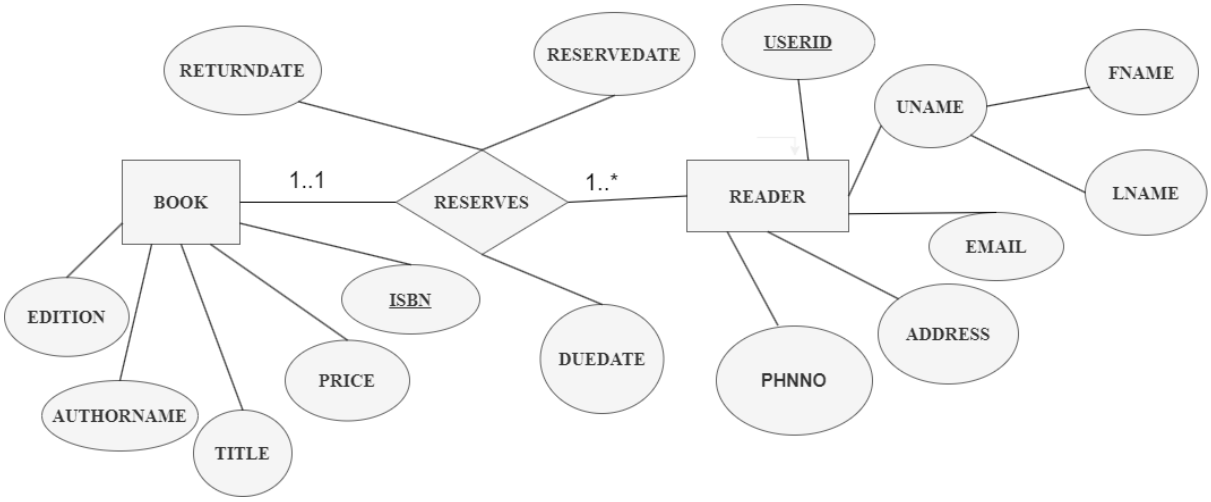


Fig 4.6 Reserves relation

UNF:
UserID,Email,UName,FName,LName,PhnNo,Address,Edition,AuthorName,Title,Price, ISBN

1NF:
UserID,Email,UName,FName,LName,PhnNo,Address,Edition,AuthorName,Title, Price,ISBN

2NF: 1. UserID(PK),Email,FName,LName,PhnNo,Address,ISBN(FK)

2. ISBN(PK),Edition,AuthorName,Title,Price

3NF: As same as 2NF

5. FINALIZATION:

1. Sid(Pk),Sname [Staff]
2. Regno(Pk),Readerid,Bookno,Riinfo [Report]
3. Sid(Pk),Regno(Fk) [Generate]
4. Userid(Pk),Email,Fname,Lname,Phnno,Address,Isbn(Fk),Reservedate, Returndate,Duedate [Reserve]
5. Isbn(Pk),Edition,Authorname,Title,Price [Book]
6. Loginid(Pk),Pass,Sid(Fk) [Track]
7. Pid(Pk),Pname,Pyear [Publisher]
8. Pid(Pk),Isbn(Fk) [Publish]
9. Sname,Sid (Pk), Userid (Fk) [Record]
10. Userid (Pk),Email,Phnno,Address,Fname,Lname [Reader]
11. Sname, Sid (Pk),Isbn (Fk) [Maintain]

6. TABLE CREATION :-

Table creation command and description of the table Staff

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10

```
create table staff(sid number(5) primary key,sname varchar(20))

describe staff
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **STAFF**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STAFF	SID	Number	-	5	0	1	-	-	-
	SNAME	Varchar2	20	-	-	-	✓	-	-

1 - 2

Fig 6.1 Creation and description of table staff

Table creation command and description of the table Report

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save Ru

```
create table report(regno number(5) primary key,readerid number(5),bookno number(5),riinfo
varchar(20))

describe report
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **REPORT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
REPORT	REGNO	Number	-	5	0	1	-	-	-
	READERID	Number	-	5	0	-	✓	-	-
	BOOKNO	Number	-	5	0	-	✓	-	-
	RIINFO	Varchar2	20	-	-	-	✓	-	-

Fig 6.2 Creation and description of table Report

Table creation command and description of the table generate

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save

```
create table generate(sid number(5) primary key,regno number(5),constraint rfk foreign
key(regno) references report(regno))

describe generate
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **GENERATE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
GENERATE	SID	Number	-	5	0	1	-	-	-
	REGNO	Number	-	5	0	-	✓	-	-

Fig 6.3 Creation and description of table generate

Table creation command and description of the table book

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
create table book(isbn number(5) primary key,edition varchar(20),authorname varchar(20),title
varchar(20),price number(5))


describe book
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **BOOK**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BOOK	ISBN	Number	-	5	0	1	-	-	-
	EDITION	Varchar2	20	-	-	-	✓	-	-
	AUTHORNAME	Varchar2	20	-	-	-	✓	-	-
	TITLE	Varchar2	20	-	-	-	✓	-	-
	PRICE	Number	-	5	0	-	✓	-	-

Fig 6.4 Creation and description of table book

Table creation command and description of the table reserve

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save

```
create table reserve(userid number(5) primary key,email varchar(20),fname varchar(20),lname
varchar(20),phnno number(11),address varchar(20),reservedate date,returndate date,duedate
date,isbn number(5),constraint bfk foreign key(isbn) references book(isbn))

describe reserve
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **RESERVE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RESERVE	USERID	Number	-	5	0	1	-	-	-
	EMAIL	Varchar2	20	-	-	-	✓	-	-
	FNAME	Varchar2	20	-	-	-	✓	-	-
	LNAME	Varchar2	20	-	-	-	✓	-	-
	PHNNO	Number	-	11	0	-	✓	-	-
	ADDRESS	Varchar2	20	-	-	-	✓	-	-
	RESERVEDATE	Date	7	-	-	-	✓	-	-
	RETURNDATE	Date	7	-	-	-	✓	-	-
	DUEDATE	Date	7	-	-	-	✓	-	-
	ISBN	Number	-	5	0	-	✓	-	-

Fig 6.5 Creation and description of table reserve

Table creation command and description of the table track

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save

```
create table track(loginid number(5) primary key, pass varchar(20) , sid number(5),
constraint sfk foreign key(sid) references staff(sid))

describe track
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **TRACK**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
TRACK	LOGINID	Number	-	5	0	1	-	-	-
	PASS	Varchar2	20	-	-	-	✓	-	-
	SID	Number	-	5	0	-	✓	-	-

Fig 6.6 Creation and description of table track

Table creation command and description of the table publisher

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Sav

```
create table publisher(pid number(5) primary key,pname varchar(20),pyear number(5))

describe publisher
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **PUBLISHER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PUBLISHER	PID	Number	-	5	0	1	-	-	-
	PNAME	Varchar2	20	-	-	-	✓	-	-
	PYEAR	Number	-	5	0	-	✓	-	-

Fig 6.7 Creation and description of table publisher

Table creation command and description of the table publish

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save

```
create table publish(pid number(5) primary key,isbn number(5),constraint pfk foreign
key(isbn) references book(isbn))

describe publish
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **PUBLISH**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PUBLISH	PID	Number	-	5	0	1	-	-	-
	ISBN	Number	-	5	0	-	✓	-	-

Fig6.8 Creation and description of table publish

Table creation command and description of the table reader

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
create table reader(userid number(5) primary key,email varchar(20),phnno number(11),address
varchar(20),fname varchar(20),lname varchar(20))

describe reader
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **READER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
READER	USERID	Number	-	5	0	1	-	-	-
	EMAIL	Varchar2	20	-	-	-	✓	-	-
	PHNNO	Number	-	11	0	-	✓	-	-
	ADDRESS	Varchar2	20	-	-	-	✓	-	-
	FNAME	Varchar2	20	-	-	-	✓	-	-
	LNAME	Varchar2	20	-	-	-	✓	-	-

Fig 6.9 Creation and description of table reader

Table creation command and description of the table record

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
create table record(sname varchar(20),sid number(5) primary key,userid number(5),constraint
ufk foreign key(userid) references reader(userid))

describe record
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **RECORD**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RECORD	SNAME	Varchar2	20	-	-	-	✓	-	-
	SID	Number	-	5	0	1	-	-	-
	USERID	Number	-	5	0	-	✓	-	-

Fig 6.10 Creation and description of table record

Table creation command and description of the table maintain

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save Ru

```
create table maintain(sname varchar(20),sid number(5) primary key,isbn number(5),constraint
mfk foreign key(isbn) references book(isbn))

describe maintain
```

Object Type **TABLE** Object **MAINTAIN**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MAINTAIN	SNAME	Varchar2	20	-	-	-	✓	-	-
	SID	Number	-	5	0	1	-	-	-
	ISBN	Number	-	5	0	-	✓	-	-

Fig 6.11 Creation and description of table maintain

7. VALUE INSERTION :-

- All COLUMMNS AND COMPONENTS OF THE TABLE STAFF-

Results Explain Describe Saved SQL History

SID	SNAME
1111	PRANTO
2222	AMIT
3333	ALAVI
4444	RIASAD

4 rows returned in 0.00 seconds CSV Export

Fig 7.1. Insertion of table staff

❖ All COLUMMNS AND COMPONENTS OF THE TABLE REPORT

Results Explain Describe Saved SQL History

REGNO	READERID	BOOKNO	RIINFO
101	12345	99	ISSUED
202	56789	88	ISSUED
303	23456	77	RETURNED
404	34567	66	RETURNED

4 rows returned in 0.00 seconds CSV Export

-

Fig 7.2. Insertion of table report

❖ All COLUMMNS AND COMPONENTS OF THE TABLE GENERATE

Results Explain Describe Saved SQL History

SID	REGNO
1111	101
2222	202
3333	303
4444	404

4 rows returned in 0.00 seconds CSV Export

-

Fig 7.3. Insertion of table generate

❖ All COLUMMNS AND COMPONENTS OF THE TABLE RESERVE-

Results Explain Describe Saved SQL History

USERID	EMAIL	FNAME	LNAME	PHNNO	ADDRESS	RESERVEDATE	RETURNDATE	DUEDATE	ISBN
9911	faruq@gmail.com	john	kabir	1630500311	UTTARA	20-JUN-23	30-JUN-23	03-JUL-23	1
7711	dip@gmail.com	datta	dip	1630508585	KHILKHET	01-AUG-23	10-AUG-23	20-AUG-23	3
6611	chowdhury@gmail.com	hasan	huq	1630503434	KURIL	01-SEP-20	10-SEP-20	20-SEP-20	4
8811	fazleh@gmail.com	fazle	rabbi	1630500624	KURIL	10-JUN-23	20-JUN-23	30-JUN-23	2

Fig 7.4. Insertion of table reserve

❖ All COLUMMNS AND COMPONENTS OF THE TABLE BOOK-

Results Explain Describe Saved SQL History

ISBN	EDITION	AUTHORNAME	TITLE	PRICE
1	3rd	AMIT IQBAL	AMITS BIOGRAPHY	2000
2	1st	PRANTO TAGORE	JOY OF HEAVEN	3000
3	8th	RIASAD THAKUR	INFINITE HAPPINESS	2500
4	9th	ALAVI CHAKROBARTY	TALK OF EYES	1000

4 rows returned in 0.00 seconds CSV Export

Fig 7.5. Insertion of table book

❖ All COLUMMNS AND COMPONENTS OF THE TABLE TRACK-

Results Explain Describe Saved SQL History

LOGINID	PASS	SID
2323	@seeme@	1111
4545	#hello*	2222
5656	(rock 20)	3333
7878	^345%	4444

4 rows returned in 0.00 seconds CSV Export

Fig 7.6. Insertion of table track

❖ All COLUMMNS AND COMPONENTS OF THE TABLE PUBLISHER-

Results Explain Describe Saved SQL History

PID	PNAME	PYEAR
22488	hasan	2011
22688	husaiyn	2002
22988	karim	2003
22788	rafiq	1999

4 rows returned in 0.00 seconds CSV Export

Fig 7.7. Insertion of table publisher

❖ All COLUMMNS AND COMPONENTS OF THE TABLE PUBLISH-

Results Explain Describe Saved SQL History

PID	ISBN
22488	1
22688	2
22988	3
22788	4

4 rows returned in 0.00 seconds CSV Export

Fig 7.8. Insertion of table publish.

❖ All COLUMMNS AND COMPONENTS OF THE TABLE RECORD-

Results Explain Describe Saved SQL History

SNAME	SID	USERID
PRANTO	1111	9911
AMIT	2222	8811
ALAVI	3333	7711
RIASAD	4444	6611

4 rows returned in 0.02 seconds CSV Export

Fig 7.9. Insertion of table record

❖ All COLUMMNS AND COMPONENTS OF THE TABLE READER-

Results Explain Describe Saved SQL History

USERID	EMAIL	PHNNO	ADDRESS	FNAME	LNAME
9911	faruq@gmail.com	1630500311	UTTARA	john	kabir
7711	dip@gmail.com	1630508585	KHILKHET	datta	dip
6611	chowdhury@gmail.com	1630503434	KURIL	hasan	huq
8811	fazleh@gmail.com	1630500624	KURIL	fazle	rabbi

4 rows returned in 0.02 seconds CSV Export

Fig 7.10. Insertion of table reader

❖ All COLUMMNS AND COMPONENTS OF THE TABLE MAINTAIN-

Results Explain Describe Saved SQL History

SNAME	SID	ISBN
PRANTO	1111	1
AMIT	2222	2
ALAVI	3333	3
RIASAD	4444	4

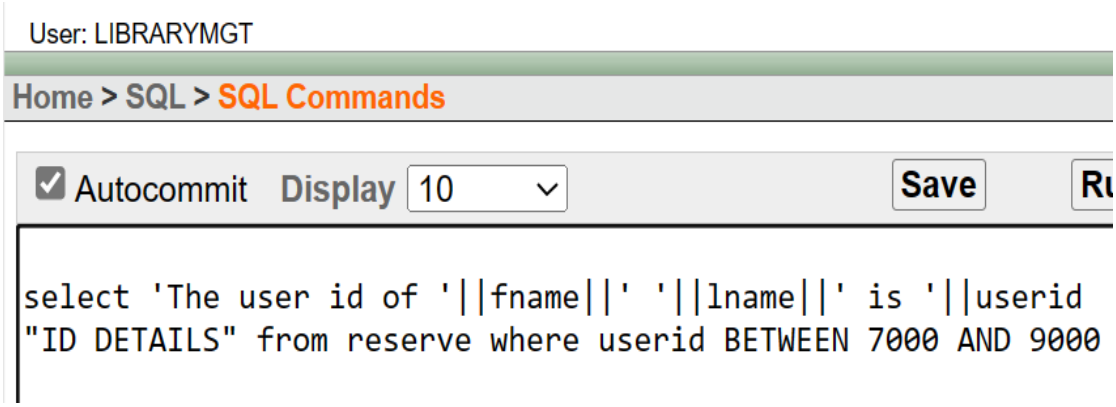
4 rows returned in 0.00 seconds CSV Export

Fig 7.11. Insertion of table maintain.

8. QUERY TEST :-

8.1 SIMPLE QUERY-

QUESTION- Write a query to show “The user id of fname lname is userid” where user id is in the range between 7000 and 9000



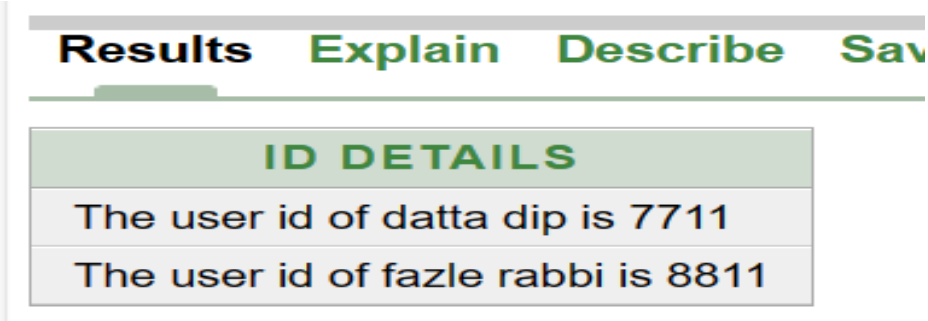


Fig 8.1 Simple query

8.2 Single row function query

QUESTION- Write a query to manipulate returndate to 'Day-Month-Year' using a single row function

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10

```
select to_char(returndate, 'fmDay Month Year') from reserve
```

Results Explain Describe Saved SQL History

TO_CHAR(RETURNDATE,'FMDAYMONTHYEAR')
Friday June Twenty Twenty-Three
Thursday August Twenty Twenty-Three
Thursday September Twenty Twenty
Tuesday June Twenty Twenty-Three

Fig 8.2 Single row function query

8.3 Aggregate function query

QUESTION- Write a query to show the minimum,average,maximum of the prices of book of table book and the number of editions of the book

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10

```
select sum(price),avg(price),min(price),count(edition) from book
```

Results Explain Describe Saved SQL History

SUM(PRICE)	AVG(PRICE)	MIN(PRICE)	COUNT(EDITION)
8500	2125	1000	4

Fig 8.3 Aggregate function query

8.4 Single row subquery

QUESTION- Write a query to show the email,phnno and fname from reader table where email is from the person with the address khilkhet and userid is greater than from the person with lname huq

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
select email,phnno,fname from reader where email=(select
email from reader where address='KHILKHET')and USERID>(select
userid from reader where lname='huq')
```

Results Explain Describe Saved SQL

EMAIL	PHNNO	FNAME
dip@gmail.com	1630508585	datta

Fig 8.4 Single row subquery

8.5 Multiple row subquery

QUESTION- Write a query to show regno,bookno from table report where riinfo is issued

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
select regno,bookno from report where riinfo=all(select
riinfo from report where riinfo='ISSUED')
```

Results Explain Describe

REGNO	BOOKNO
101	99
202	88

Fig 8.5 Multiple row subquery

8.6.1 Equijoin

QUESTION- Write a query to join sid ,sname of staff table and loginid,pass of track table with a valid condition

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save

```
select staff.sid,staff.sname,track.loginid,track.pass from
staff,track where staff.sid=track.sid
```

Results Explain Describe Saved SQL H

SID	SNAME	LOGINID	PASS
1111	PRANTO	2323	@seeme@
2222	AMIT	4545	#hello*
3333	ALAVI	5656	(rock 20)
4444	RIASAD	7878	^345%

Fig 8.6.1 Equijoin

8.6.2 Self join-

QUESTION- Write a query to show "pname publishes in pyear" with self join putting pname and pyear in different object aliases and pyear being the identitcal column between them

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save

```
select pub.pname|| ' publishes in ' ||ye.pyear as "PUBLISH
DETAILS" from publisher pub,publisher ye where
pub.pyear=ye.pyear
```

Results Explain Describe

PUBLISH DETAILS
hasan publishes in 2011
husaiyn publishes in 2002
karim publishes in 2003
rafiq publishes in 1999

Fig 8.6.2 Self join

8.7.1 Simple view-

Question- Create a simple view to show userid,fname and email of reader table

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
create view vureader as select userid,fname,email from reader

select * from vureader
```

Results Explain Describe Saved SQL His

USERID	FNAME	EMAIL
9911	john	faruq@gmail.com
7711	datta	dip@gmail.com
6611	hasan	chowdhury@gmail.com
8811	fazle	fazleh@gmail.com

Fig 8.7.1 Simple view

8.7.2 Complex view-

QUESTION- Create a complex view which contains sname,sid,authorname,isbn from table maintain and book where isbn column matches between the two table

User: LIBRARYMGT

Home > SQL > SQL Commands

Autocommit Display 10 Save

```
create view vumaintain as select
b.isbn,b.authorname,m.sname,m.sid from book b,maintain m
where b.isbn=m.isbn

select * from vumaintain
```

Results Explain Describe Saved SQL History

ISBN	AUTHORNAME	SNAME	SID
1	AMIT IQBAL	PRANTO	1111
2	PRANTO TAGORE	AMIT	2222
3	RIASAD THAKUR	ALAVI	3333
4	ALAVI CHAKROBARTY	RIASAD	4444

Fig 8.7.2 Complex view

9.1 DATABASE CONNECTION:-

(IRTISAM FARUQUI ALAVI, 22-48863-3)

1. The mysql java connector(jar file) is installed at first . For that, the mysql java connector maven containing jar file of version 8.0.28 is downloaded.
2. The xampp apache mariadb perl php mysql xampp server id downloaded which contains many servers.Among that the apache and mysql server is started and admin panel of mysql server is opened
3. After that in that panel a database system is created namely "librarymgt",and a table namely "publisher" which contains three columns . The subsequent values are then inserted into the table comprising four rows in total

   The table in mysql-

PID	PNAME	PYEAR
22488	hasan	2011
22688	husaiyn	2002
22988	karim	2003
22788	rafiq	1999

Fig 9.1.1 Output of the table from the mysql server

4. Then an IDE is downloaded and installed namely "Apache Netbeans IDE20".Inside it, the jar file is added in the library section of the new project created in the IDE.
5. After than a DB connection code is written iniside the "LMGT" class of the project. The basis of the code comprises of the following criteria's-

   - Register Driver- In the provided code, the line Class.forName("com.mysql.cj.jdbc.Driver"); is used to dynamically load the MySQL JDBC (Java Database Connectivity) driver. In JDBC, drivers are used to establish a connection between a Java application and a database. Loading the driver is necessary to register it with the DriverManager, which allows the application to use the specified database driver.

   - Connection of DB- The DriverManager is used to establish a connection to a database by loading the appropriate driver and creating a connection.The connection interfeace represents a connection to a database.It provides methods for creating statements.The connection object is obtained from the

DriverManager by calling the getConnection method with a URL,username and password.

- Statement- The Statement interface represents a SQL statement that can executed against a database.Statement objects are created using the createStatement method of a Connection

- Execution of the Query() in the Statement- The ResultSet interface represents the result set of a SQK query. It provides methods for retrieving data from the result set,iterating through rows, and accessing column values.ResultSet objects are obtained by executing a query on a Statement

- Connection close()-The connection.close() method in Java is used to close a database connection. Closing a connection is important for memory management, resource management because its important to release the resources when they are no longer needed

The whole code-

```
import java.sql.*;
public class LMGT
{
    public static void main(String[] args){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection connection= DriverManager.getConnection("jdbc:mysql://localhost:3306/librarymgt","root", "");
            System.out.println("connected");
            Statement statement;
            statement =connection.createStatement();
            ResultSet resultset;
            resultset = statement.executeQuery("select * from publisher");

            while(resultset.next()){
                System.out.println(" PID = " + resultset.getInt(1)+" PNAME = "+ resultset.getString(2)+ ", PYEAR = "+resultset.getInt(3));}
            connection.close();
        }catch (Exception s){
            System.out.println(s);
        }
    }
}
```

Fig 9.1.2 Java code to connect mysql database and to show the output

The output is –

```
Output - LMGT (run)
run:
connected
 PID = 22488 PNAME = hasan, PYEAR = 2011
 PID = 22688 PNAME = husaiyn, PYEAR = 2002
 PID = 22988 PNAME = karim, PYEAR = 2003
 PID = 22788 PNAME = rafiq, PYEAR = 1999
BUILD SUCCESSFUL (total time: 0 seconds)
```

Fig 9.1.3 Output of the query from the code

9.2 (MD. FAZLEH RABBI YAHIA PRANTO, 22-49575-3)

To make database connection work we first need to have few things installed or downloaded on our device if we do not have it already.

1. You will have to have jdk and netbeans installed on your device first to work on it.
2. Mysql connector maven
   - When we search it on web browser we will see a search result naming maven respository. Click on the link we will have to click the 8.0.28 version of the file and download the jar file.

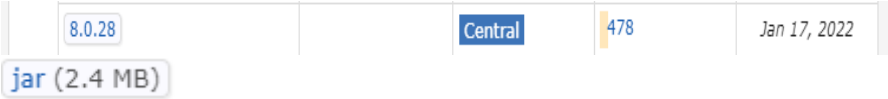


Fig 9.2.1 Output of the downloaded section

3. Xampp apcahe mariadb perl php
   - It will show a search result naming Apache Friends and click the download link depending on the OS you are using it should start downloading automatically.

     Install the app (have to install it on your default drive if it is in C: drive depending on the device, or it will not work correctly) after installation start apache and start mysql. you will have click admin on mysql.when you click on the admin button a webpage will open on your default browser.go to database there and create your management system for example my project is on librarymgt.create a table of your choice I created 'staff'
     Here is my mysql table-

sid	sname
1111	PRANTO
2222	AMIT
3333	ALAVI
4444	RIASAD

Fig 9.2.2 Output of the table from mysql server

4. The IDE we chose 'netbeans' open it and create a new project and add the jar file of mysql connector on the library section.
5. After that a DB connection code is written inside the "LMGT" class of the project. The basis of the code comprises of the following criteria's-
   - Register Device
   - Connection of DB
   - Create Satement
   - Execution of query() in the Statement
   - Connection close()

The whole code-

```
package lmgt;
import java.sql.*;
/**
 *
 * @author Farhana
 */
public class Lmgt {

  public static void main(String[] args){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection connection= DriverManager.getConnection("jdbc:mysql://localhost:3306/librarymgt","root", "");
            System.out.println("connected");
            Statement statement;
            statement =connection.createStatement();
            ResultSet resultset;
            resultset = statement.executeQuery("select * from staff");
            while(resultset.next()){
                System.out.println(" SID = " + resultset.getInt(1)+" SNAME = "+ resultset.getString(2));
            }
        connection.close();
        }catch (Exception s){
            System.out.println(s);
        }
    }
}
```

Fig 9.2.3 Java code to connect to DB connection and to show output

The output is-

```
Output - lmgt (run)
run:
connected
 SID = 1111 SNAME = PRANTO
 SID = 2222 SNAME = AMIT
 SID = 3333 SNAME = ALAVI
 SID = 4444 SNAME = RIASAD
BUILD SUCCESSFUL (total time: 3 seconds)
```

Fig 9.2.4 Output of the query of the java code

CONCLUSION-

The summary of the whole report is about a typical library database management system which comprises the usual processes and relevant components needed while making a database system . ER diagram was based on the case study which is followed by normalization, finalization . Table was created afterwards in the oracle database platfrom ,subsequently inserting values into them.A query test is performed to examine the database system. At last not but the least, a DB connection was established using the mysql server and netbeans ide of the table publisher. In conclusion, this report serves as a valuable resource for the people involved in education feild