



EAST WEST UNIVERSITY

Library Management System

Course Name: Database Systems

Course: CSE-301

Section: 01

Group No: 01

Submitted By:

Tithir Mahmud Bakshi	2016-1-60-095	33%
Snehashis Ghosh Pial	2017-2-60-163	34%
Afrina Mustofa	2018-1-60-147	33%

Submitted To:

Md Mostofa Kamal Rasel

Assistant Professor,

Department of Computer Science and Engineering.

Submitted on: 9th January 2022

LIBRARY MANAGEMENT SYSTEM

Project Description:

A student and faculty can issue books. Different limits for the number of books a student and teacher can issue. Also, the number of days will be distinct in the case of students and teachers for issue any book. Each book will have different ID. Also, each book of the same name and same author (but the number of copies) will have different ID. Entry of all the book will be done(report), who issue that book and when and also duration. Detail of Fine(when the book is not returned at a time) is also stored. Both admin and user can send message to each other.

Objective:

In this project, our goal is to develop a robust system which can store informations and manage gently the library management system. Its main functionality is to manage the details of address, member, books, issued books, track the users etc. This project is designed to give a user friendly service to borrow and return book. This system is designed to manage library management system electronically which is easier and more reliable. It is built on administrative end, where admin has all the power and access to everything. Admin can handle this management very easily.

- More Accuracy
- Speed improvement
- Data inconsistency
- Better error handling
- Integrity

Entity sets:

- **Admin:** Admin keeps the information about Admin ID, name, username, password, picture, phone no, email of Admins. The administrator of the Admin manages reports, keep tracks of and send or receive messages from students and teachers and maintains books.

Admin
Admin_id_no
Name
first_name
last_name
username
password
{phone}
email

Prepared by: Afrina Mustofa

- **Report:** Report keeps information about delay fine, username, book ID, return date for all books and is managed by Admin.

Rerort
Delay_fine
Username
Book_id
Return_date

Prepared by: Afrina Mustofa

- **Student:** Student keeps information about the student ID, name, Username, password, phone no and email of all the students. Students have the permission to borrow or return books and send or receive text from the Admin.

Student
Student_id_no
Name
first_name
last_name
Username
password
{phone}
email

Prepared by: Afrina Mustofa

- **Book:** Book keeps information about the book ID, title, author's name, book type, edition which is maintained by the Admin. Teachers and the students has permission to borrow books that are available here.

Book
<u>Book_id</u> Title Authors_name Book_type {Edition}

Prepared by: Afrina Mustofa

- **Teacher:** Teacher keeps information about the teachers' ID, name, username, password, phone no, picture and email of all the teachers. Teachers have the permission to borrow or return books and send or receive text from the Admin.

Teacher
Teacher_id_no Name first_name last_name Username password picture {phone} email

Prepared by: Afrina Mustofa

- **Issue book:** Issue_book can keep the record that which user is borrow the book and which user request is approved.

Issue_book
username book_id approve issue_date return_date

Prepared by: Snehashis Ghosh Pial

Cardinality constraints and participation:

- **Admin – Report:**

- **Cardinality constraint:** The cardinality constraint from the “Admin” entity set to the “Report” entity set is one–to–many. This means an Admin can generate more than one report but one Report cannot be generated by many Admins.
- **Participation:** An admin may generate a report or not. That’s why the entities of the “Admin” entity set participate partially in the “Manages” relationship. On the other side, a Report must be generated by any Admin. That’s why the entities of the “Report” entity set participate fully in the “Manages” relationship.

Prepared by: Tithir Mahmud Bakshi

- **Admin – Student:**

- **Cardinality constraint:** The cardinality constraint from the “Admin” entity set to the “Student” entity set is many–to–many. This means an Admin can keep track of more than one student and a student can interact with many Admins.
- **Participation:** An Admin may or may not keep the records of students. That’s why the entities of the “Admin” entity set participate partially in the “Keep tracks of” relationship. On the other side, a student must interact with at least one Admin. That’s why the entities of the “Student” entity set participate fully in the “Keep tracks of” relationship.

Prepared by: Tithir Mahmud Bakshi

- **Admin – Book:**

- **Cardinality constraint:** The cardinality constraint from the “Admin” entity set to the “Book” entity set is many–to–many. This means many Admin can maintain books and a Book can be maintained by many Admin.
- **Participation:** An admin may or may not maintain a book. That’s why the entities of the “Admin” entity set participate partially in the “Maintains” relationship. On the other side, a book must be maintained by an Admin. That’s why the entities of the “Book” entity set participate fully in the “Maintains” relationship.

Prepared by: Snehashis Ghosh Pial

- **Admin – Teacher:**

- **Cardinality constraint:** The cardinality constraint from the “Admin” entity set to the “Teacher” entity set is many–to–many. This means many Admin can keep track of a teacher and many Teachers can be kept track of by an Admin.
- **Participation:** An admin may or may not keep track of teachers. That’s why the entities of the “Admin” entity set participate partially in the “Keep track of” relationship. On the other side, a teacher must interact with an Admin. That’s why the entities of the “Teacher” entity set participate fully in the “Keep track of” relationship.

Prepared by: Snehashis Ghosh Pial

- **Book – Teacher:**

- **Cardinality constraint:** The cardinality constraint from the “Teacher” entity set to the “Book” entity set is many–to–many. This means a teacher must borrow a book but a book may or may not be borrowed by a teacher.
- **Participation:** A student must borrow/return a book. That’s why the entities of the “Teacher” entity set participate fully in the “Borrow/Return” relationship. On the other side, a book may or may not be borrowed by a teacher. That’s why the entities of the “Book” entity set participate partially in the “Borrow/return” relationship.

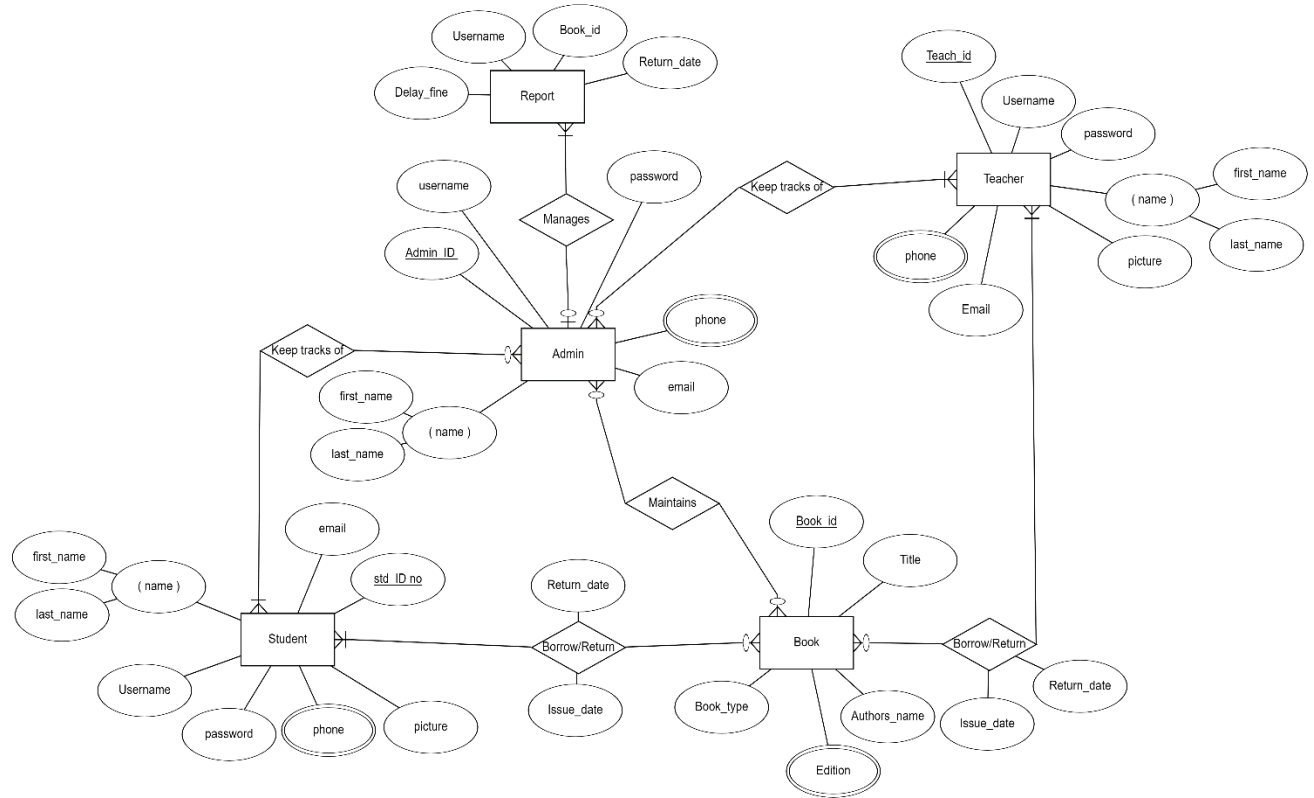
Prepared by: Tithir Mahmud Bakshi

- **Student – Book:**

- **Cardinality constraint:** The cardinality constraint from the “Student” entity set to the “Book” entity set is many–to–many. This means a student must borrow a book but a book may or may not be borrowed by a student.
- **Participation:** A student must borrow/return a book. That’s why the entities of the “Student” entity set participate fully in the “Borrow/Return” relationship. On the other side, a book may or may not be borrowed by a student. That’s why the entities of the “Book” entity set participate partially in the “Borrow/return” relationship.

Prepared by: Tithir Mahmud Bakshi

ER Diagram:



Prepared by: Snehashis Ghosh Pail, Tithir Mahmud Bakshi, Afrina Mustofa

Relational Schema:

Prepared by: Snehashis Ghosh Pial

Query2: Find the information of borrowed books?

```
select
student.Username,student.first_name,student.last_name,book.book_id,title,authors_name,edition
,issue_date,return_date

from student inner join issue_book ON student.Username=issue_book.username inner join book
ON issue_book.book_id=book.book_id

where issue_book.approve='Approved'

order by return_date ASC;
```

Screenshot2,

+ Options

Username	first_name	last_name	book_id	title	authors_name	edition	issue_date	return_date	▲ 1
2017-2-60-163	Snehashis Ghosh	Pial	2	Introduction to Computer Science	G.Michael Schneider	1st	2021-11-18	2021-12-25	
2017-2-60-160	Jashedul Islam	Jusef	3	Introduction to finance	Ronald W. Melicher	1st	2021-12-18	2021-12-28	

☐ Show all | Number of rows: 25 ▼ | Filter rows: Search this table | Sort by key: None ▼

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

Prepared by: Snehashis Ghosh Pial

Query3: Find the fine for delay book submission?

```
with

temp as(select borrower_username,book_id,title,Timestampdiff(day,(select returns_date from
report where return_status="EXPIRED"), curdate()) as delay_day


from report natural join book

where return_status="EXPIRED")

select borrower_username,book_id,title,delay_day, delay_day*50 as delay_fine
```

from temp;

screenshot 3,

 Showing rows 0 - 0 (1 total, Query took 0.0042 seconds.)

```
with temp as(select borrower_username,book_id,title,Timestampdiff(day,(select returns_date from report where
return_status="EXPIRED"), curdate())) as delay_day from report natural join book where return_status="EXPIRED") select
borrower_username,book_id,title,delay_day, delay_day*50 as delay_fine from temp
```

[Edit inline] [Edit] [Create PHP code]

+ Options

borrower_username	book_id	title	delay_day	delay_fine
2017-2-60-160	3	Introduction to finance	13	650

Query results operations

Prepared by: Snehashis Ghosh Pial

Query4: Find the information which admin interact with which user?

Select

report.admin_username,admin.first_name,admin.last_name,borrower_username,book_id,book.title

from admin inner join report ON admin.Admin_ID=report.Admin_username natural join book;

screenshot 4,

```
Select report.admin_username,admin.first_name,admin.last_name,borrower_username,book_id,book.title from admin inner join report ON
admin.Admin_ID=report.Admin_username natural join book
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

admin_username	first_name	last_name	borrower_username	book_id	title
2015-2-13-151	Selina	Biswas	2017-2-60-160	3	Introduction to finance
2015-2-70-150	Santanu	Sarkar	2017-2-60-160	3	Introduction to finance
2015-2-15-153	Md. Tarikul	Islam	2017-2-60-163	2	Introduction to Computer Science
2015-2-15-154	Debasish	Ghosh	2016-1-60-095	3	Introduction to finance

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Prepared by: Tithir Mahmud Bakshi

Query5: Find each book type wise number of student who borrow the specific type of book?

with

temp as(select book_id,title,book_type,username

from issue_book natural join book)

select book_type,count(username) as num_of_borrower

from temp

group by book_type;

screenshot 5,

Showing rows 0 - 2 (3 total, Query took 0.0042 seconds.)

```
with temp as(select book_id,title,book_type,username from issue_book natural join book) select book_type,count(username) as
num_of_borrower from temp group by book_type
```

[Edit inline] [Edit] [Create PHP code]

+ Options

book_type	num_of_borrower
Bussiness	2
History	1
Science	1

Query results operations

Prepared by: Snehashis Ghosh Pial

Query6: Find not approved request for teacher?

select

teacher.Username,teacher.first_name,teacher.last_name,book.book_id,title,authors_name,edition
,status,quantity

from teacher inner join issue_book ON teacher.teach_id=issue_book.username inner join book
ON issue_book.book_id=book.book_id

where issue_book.approve='APPROVED';

screenshot 6,

Showing rows 0 - 0 (1 total, Query took 0.0052 seconds.)

```
select teacher.Username,teacher.first_name,teacher.last_name,book.book_id,title,authors_name,edition,status,quantity from teacher
inner join issue_book ON teacher.teach_id=issue_book.username inner join book ON issue_book.book_id=book.book_id where
issue_book.approve='APPROVED'
```

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25 | Filter rows:

+ Options

Username	first_name	last_name	book_id	title	authors_name	edition	status	quantity
Wasifreza	Wasif	Reza	4	Politico-Military Strategy of the Bangladesh Liber...	Guru Saday Batabyal	1st	Available	3

☐ Show all | Number of rows: 25 | Filter rows:

Prepared by: Tithir Mahmud Bakshi

Update & Delete Queries:

Query1: Delete Science type book,

delete from book

where book_type in(select book_type

from book

where book_type="Science");

Prepared by: Snehashis Ghosh Pial

Query2: Detete student from the report who return the book in due time,

delete from report

where borrower_username not in(select borrower_username
from report
where return_status="EXPIRED");

Prepared by: Tithir Mahmud Bakshi

Query3: Update the return time is expired or not,

update issue_book set approve="EXPIRED" where return_date<(select CURRENT_DATE())
and approve="Approved" LIMIT 10;

Prepared by: Snehashis Ghosh Pial

Query4: Update the edition of the economic type book,

update book set edition="3rd"
where book_id in(select book_id from book
where book_type="Economic");

Prepared by: Snehashis Ghosh Pial

Query5: Update the not approved request to approve,

update issue_book set approve="Approved",issue_date="2021-12-18",

```
return_date="2021-12-28"
```

```
where username in(select username from issue_book  
                    where username="2016-1-60-095" and book_id=3);
```

Prepared by: Afrina Mustofa

View and Materialized View:

As our database is transactional view can provide the better performance. We can hide the user confidential data like password, contact information from intruder. We also hide the delay_fine of the user.

VIEW
















Hide the confidential information from the other user,






```
create view student_info as
```

```
select std_id_no,first_name,last_name,username  
from student;
```

Execute: select * from student_info;

+ Options

				std_id_no	first_name	last_name	username
<input type="checkbox"/>	 Edit	 Copy	 Delete	2016-1-60-095	Tithir Mahmud	Bakshi	2016-1-60-095
<input type="checkbox"/>	 Edit	 Copy	 Delete	2016-2-60-095	Hamza Mahmud	Takir	2016-2-60-095
<input type="checkbox"/>	 Edit	 Copy	 Delete	2017-2-60-160	Jashedul Islam	Jusef	2017-2-60-160
<input type="checkbox"/>	 Edit	 Copy	 Delete	2017-2-60-163	Snehashis Ghosh	Pial	2017-2-60-163
<input type="checkbox"/>	 Edit	 Copy	 Delete	2018-1-60-147	Afrina	Mustofa	2018-1-60-147

 ☐ Check all With selected:  Edit  Copy  Delete  Export

☐ Show all Number of rows:

25

 Filter rows:

Search this table

Prepared by: Snehashis Ghosh Pial

Materialized view

Hide admin information,

```
create MATERIALIZED view admin_info AS
```

```
select Admin_id,first_name,last_name,email  
from admin;
```

Execute: select * from admin_info

Prepared by: Tithir Mahmud Bakshi

Conclusion:

To conclude, our generated database is user friendly. This system allows storing the details of all data related to library. Admin, student and teacher can interact with each other very easily. We do not develop any interface. For this reason, we can not add different important features like authentication system, communication chat-box, email-verification process etc. In future, we will develop a interface using php or python so that we can represent the interaction more graphically. The implement of our database will reduce data entry time and provide readily calculated reports.