Indian Institute of Technology Gandhinagar

Library Management System

CS 432 : Project Report

Group - x

Team Members

Jayesh Bhadange	20110082
Pranav Rathod	20110143
Naval Jaggi	20110118
Harendra Khatik	20110072
Sparsh Dawra	20110203
Manpreet Singh	20110109
Sidharth Joshi	19110169
Gajanan Donge	20110061
Manish Jangir	20110107
Sankarshan	20110184
Riya Dhantoliya	20110168
Pintu Kumar Meena	19110193

Under the guidance of

Prof. Mayank Singh

CS 432: Databases

Assignment 1: DESIGNING THE DBMS

Responsibility of G1

Write a description of the database system in detail?

Q1)
The library management system is an essential database for every university. We specifically have taken account of operations handled and managed at IIT Gandhinagar. Operations like book issues, book purchases, returning, etc. This database system can be used for all such kinds of functions and may be later on, this can be scaled and in future, we can work on it to solve some problems still remaining in the library.

Stakeholder : Students, faculty, and staff are among the library's stakeholders.

Functional requirements of the database includes :

1) Allowing the library staff to add the details of new books into the database:
   (i) Book name
   (ii) Book id
   (iii) Availability
   (iv) Number of pages
   (v) Book price
   (vi) Purchase date
   (vii) Purchase ID
   (viii) Publisher Name
   (ix) Book location on the shelves.
   (x) Author Name
   (xi) Book Edition

2) The database shall allow the library staff to maintain the availability of books.
3) Online catalog for the books should have the view access for all the users regarding the details of the book including the genre, author details, location
4) Library staff will keep track of the transaction details when a user issues a book till it returns the book and also impose charges depending on the date of return and also store the penalty ID, which is paid against the charges applied on the user.
5) It should enable all the users of the university to issue and return the same books

6) Working staff's timings for in and out should be noted while working in their respective sections.
7) Library staff should have a unique system ID, and also store the specification details of that ID. For staff, we could store sections like general manager, technical section, front-desk assistant.
8) Library staff should be able to store the details regarding the details of users, which are identified by their unique ID. Users for the institute involved students, staff and faculties. Student details include the programmes, faculties details include their department, and staff could be visiting or temporary.

Q2) Stakeholders - Informatics Company

We have interacted with some people from the Library staff.

1. **Dr. TS Kumbar - Advisor (Library & Institute archives)**
2. **Mr. Tapas Kumar Das - Sr. Library Information Assistant**
3. **Ms. Pannaben Chaudhari - Assistant Librarian**

Ms. Pannaben Chaudhari

Q1 - How does the database management system of the Library work?
Q2 - What software do they use to issue and return books?
Q3 - How do they provide a unique_id for each book?
Q4 - What issues do they face during the downtime of software?
Q5 - How do they do the data entry for a library subscription?
Q6 - How does the RFid system use in the issue and return?
Q7 - What information does the tag at the side of the book provide?
Q8 - How do they manage the penalty on the books?

Mr. Tapas Kumar Das - Sr. Library Information Assistant

Q1 - What things do they ask in the users' database?
Q2 - What things do they have for the database of books?
Q3 - How do they manage the library staff in and out entry?

Dr. TS Kumbar - Advisor (Library & Institute archives)

Q1 - How do they store the database of books and users?
Q2 - In which format do they store it?
Q3 - How do they provide the information about the book publicly?
Q4 - How do the short loan and RC book system work?

Q3) Name all the entities, relationships, and attributes involved in your system.

<u>Entities and Attributes:-</u>

Serial No	***Entity Name***	***Attribute***
1	Book	BookID(Primary Key), Author, Title, Edition, Copies, AvailaibityStatus
2	Publisher	Name, PublisherID(Primary Key)
3	Issue	IssueID(Primary Key) IssueDate, DueDate
4	Return	ReturnDate, ReturnID(Primary Key)
5	Fine	FineID(Primary Key), DueAmount, PayDate
6	User	UserID(Primary Key), Name, Contact
7	Student	Program, MaxBooks
8	Faculty	Dept, Office
9	Purchase	PurchaseID(Primary Key), Price, PurchaseDate
10	Shelf	RowNo(Primary Key), SectionID(Primary Key), SectionDept

Relationship set and Entities:-

Serial No	***Relationship set***	***Entities***
1	Publishes	Publisher, Book
2	MadeBy	Issue, User
3	Pays	User, Fine
4	LateReturn	Return, Fine
5	PlacedOn	Books, Shelf
6	BookPurchase	Book, Purchase
7	IssueOfBook	Book, Issue
8	BookReturn	Book, Return
9	ReturnForIssue	Issue, Return
10	ReturnedBy	User, Return

Q4) Give examples and justification for points c to g in Design Requirements.

Relationship set

- ***Mapping cardinalities***
- ***Justification***
- ***Participation constraint***

Publishes

- One to many relationship between entities Publisher and Book.
- One publisher can print many books but the book will have a single publisher.
- Total participation constraint for entity book because all books have publisher and total participation constraint for entity publisher because at least there will be one book of publisher that is present in the library. While adding a new book in the library we will add the publisher name in the database if it is not present in the database.

MadeBy

- ➢ Many to one relationship between entities Issue and User.
- ➢ One user can issue as many books as he wants with different issue ID and one issue associated with only one user.
- ➢ Total participation constraint for entity Issue because there will be a user associated with all issues and partial participation constraint for entity User because all users will not have issue.

Pays

- ➢ One to many relationship between entities, users and fine.
- ➢ A user can have multiple fines and a specific fine can belong to only one user.
- ➢ Total participation constraint for entity Fine because there will be a user associated with all fines and partial participation constraint for entity User because all users will not have any fine.

LateReturn

- ➢ One to one relationship between entities return and fine.
- ➢ After returning the book after the due date, the user will be fined. So each issue ID will have a unique fine.
- ➢ Total participation constraint for entity Fine because there will be fine associated with return and partial participation constraint for entity Return because all returns will not have any fine as they might be returned before the due date.

PlacedOn

- ➢ Many to one relationship between entities books and shelf.
- ➢ A book belongs to a unique shelf but a shelf can have many books.
- ➢ All books will be on the shelf but there might be any shelf that will be empty so there is Total participation constraint for entity book and partial participation constraint for entity shelf.

BookPurchase

- ➢ Many to one relationship between entities books and purchase.
- ➢ In a single purchase there can be multiple books but every book will come under single purchase.
- ➢ Also all books will be purchased once and all purchase will be associated with some book, so it’s a total relationship from both sides.

IssueOfBook

- ➢ One to many relation between entities Book and issue.
- ➢ A book can be issued many times but a single issue will contain only one book.
- ➢ All issue are associated with book but all books will not be associated with issue so Total participation constraint for issue and partial participation constraint for book.

BookReturn

- ➢ One to many relation between entities Book and return.
- ➢ One book can be returned many times but each return will have a single book.
- ➢ All returns are associated with the book but all books will not be associated with return because there may be cases where a book is not issued so Total participation constraint for return and partial participation constraint for book.

ReturnForIssue

- ➢ One to one relationship between Issue and Return.
- ➢ One issue can have only one return because for a unique issue ID there will be a unique return ID.
- ➢ There will be book return associated but there may not be return for all book issue. So Total participation constraint for entity return and partial participation constraint for entity issue.

ReturnedBy

- ➢ One to many relation between entities user and return.
- ➢ One user can have multiple books to return with different issue ID's but each return with a unique return ID will have a specific user.
- ➢ There will be a user associated with all returns but every user may not have return. So Total participation constraint for entity Return and partial participation constraint for entity User.

The reason, why isn't there Many to Many cardinality for any Relationship:
Because ultimately, we have a User/ Publisher/ person to deal with the book. If we can see logically, a person can access multiple books at a time, which is Many. On the other hand, a single book can not be accessed by multiple persons at the same time.

But there can be a case when there are authors and books (both entity sets), and this relationship can have Many to Many cardinality. But we do not need this relationship in our database.

Responsibility of G2

Q1, Q2, Q3) Relationship Schema(Primary and Foreign Keys for Schema , Key constraints):

Book(BookID, Title, Edition, Copies, AvailaibityStatus, PublisherID, SectionID, RowNo, PurchaseID)

- BookID is Primary key
- PublisherID is the foreign key referencing the Publisher relstion
- (RowNo, SectionID) is the Foreign key referencing the Shelf relation
- PurchaseID is Foreign key referencing the Purchase relation
- RowNo is NOT NULL because each book must have a RowNo
- SectionID is NOT NULL because each book must have a shelf location, and each shelf has SectionID.

Book_Auth(BookID, Author)

- (BookID , Author) is the Primary key
- BookID is foreign key referencing the Book relation

Publisher(PublisherID, Name)

- PublisherID is Primary key
- Name : NOT NULL, because the name of the publisher is required.

Issue(IssueID, IssueDate, Due Date, UserID, BookID)

- IssueID is Primary key
- User ID is the Foreign key referencing the User relation
- BookID is the Foreign key referencing the Book relation
- IssueDate, Due Date can not be Null values so we have to add a NOT NULL constraint on these attributes.

Return(ReturnID, ReturnDate, FineID, BookID,IssueID,UserID)

- ReturnID is Primary key
- FineID is the Foreign key referencing the Fine relation
- BookID is the Foreign key referencing the Book relation
- IssueID is the Foreign key referencing the Issue relation
  UserID is the Foreign key referencing the User relation
- ReturnDate can not have Null values so ReturnDate should have NOT NULL constraint.

Fine(FineID, DueAmount, PayDate, UserID)

- FineID is Primary key.
- UserID is foreign key referencing the User relation.
- Constraint on DueAmount : Check(domain_value >= 0)

User(UserID, first_name,miidle_name, end_name, Contact)

- UserID is primary key
- Contact is UNIQUE
- first_name is NOT NULL because each user must have a first name.

Student(UserID, Program, MaxBooks)

- UserID is primary key
- UserID is Foreign key referencing the relation User
- MaxBooks has a CHECK where the max number of books issued can be 6.

Faculty(UserID, Dept, Office)

- UserID is primary key
- UserID is Foreign key referencing the relation User
- Dept is NOT NULL

Purchase(PurchaseID, Price, PurchaseDate)

- PurchaseID is Primary key

Shelf(SectionID, RowNo, SectionDept)

- (SectionID,RowNo) is Primary key

Q4) Mapping Cardinality:-

Serial No	***Relationship set***	***Entities***	***Mapping Cardinality***	***Participation***
1	Publishes	Publisher, Book	One to Many	Total , Total
2	MadeBy	Issue, User	Many to One	Total , Partial
3	Pays	User, Fine	One to Many	Partial , Total
4	LateReturn	Return, Fine	One to One	Partial , Total
5	PlacedOn	Book, Shelf	Many to One	Total , Partial
6	BookPurchase	Book, Purchase	Many to One	Total , Total
7	IssueOfBook	Book, Issue	One to Many	Partial , Total
8	BookReturn	Book, Return	One to Many	Partial , Total
9	ReturnForIssue	Issue, Return	One to One	Partial , Total
10	ReturnedBy	User, Return	One to Many	Partial , Total

Q5)Redundant Relationship sets or entity sets

The Relationship set Publishes is redundant. Publishes is a one to many relationship between Publisher and Book and participation of Book is total. Therefore the Relation schema of Publishes and Book can be combined into a schema of Book.

The Relationship set MadeBy is redundant. MadeBy is a one to many relationship between User and Issue and participation of Issue is total. Therefore the Relation schema of MaseBy and Issue can be combined into the schema of Issue.

The Relationship set Pays is redundant. Pays is a one to many relationship between User and Fine and participation of Fineis total. Therefore the Relation schema of Pays and Fine can be combined into the schema of Fine.

The Relationship set LateReturn is redundant. LateReturn is a one to one relationship between Return and Fine and participation of Fine is total. Therefore the Relation schema of LateReturn and Fine can be combined into the schema of Fine.

The Relationship set PalcedOn is redundant. PlacedOn is a Many to one relationship between Book and Shelf and participation of Book is total. Therefore the Relation schema of PalcedOn and Book can be combined into the schema of Book.

The Relationship set BookPurchase is redundant. BookPurchase is a Many to one relationship between Book and Purchase and participation of Book is total. Therefore the Relation schema of BookPurchase and Book can be combined into the schema of Book.

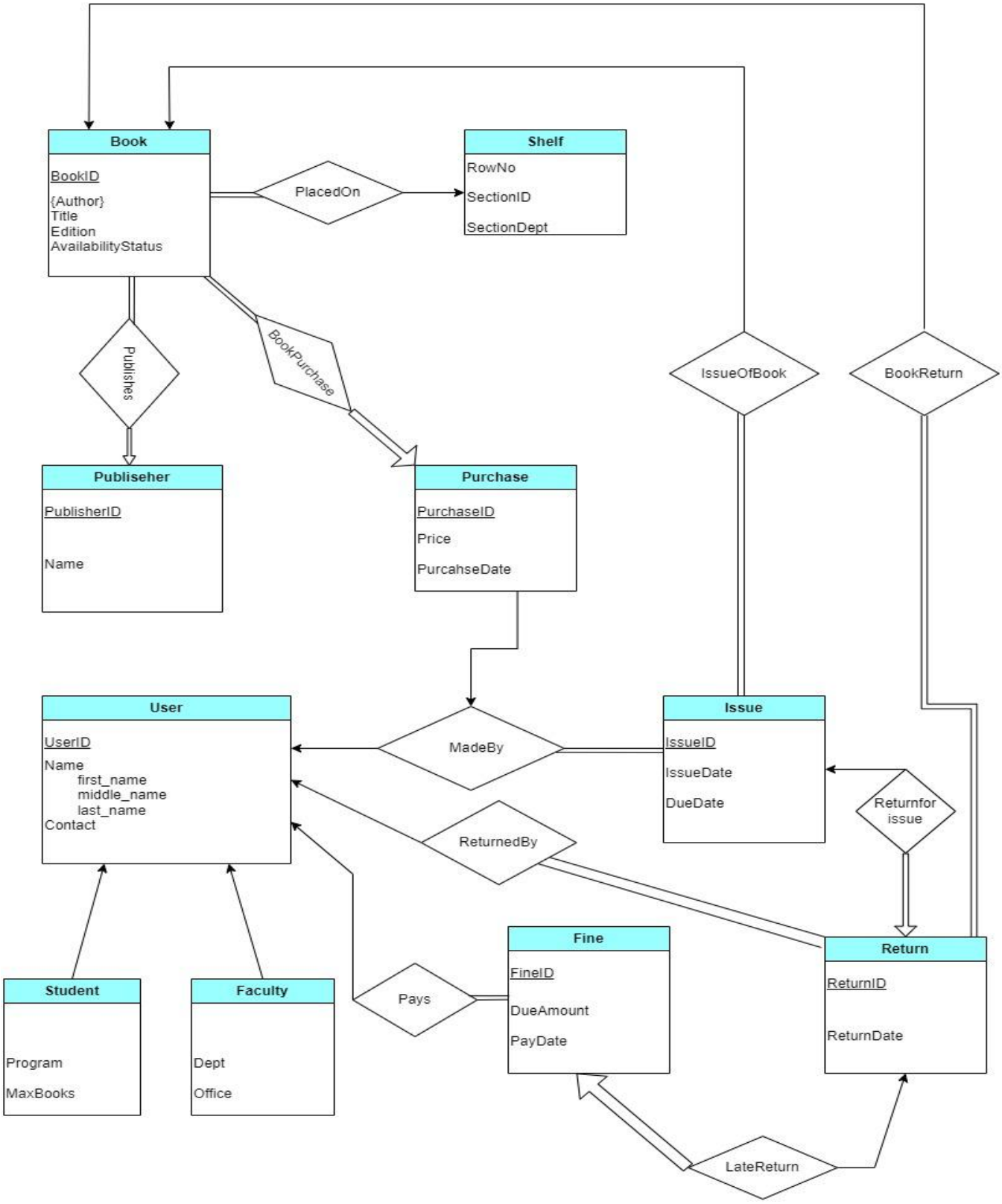
The Relationship set IssueOfBook is redundant. IssueOfBook is a One to many relationship between Book and Issue and participation of Issue is total. Therefore the Relation schema of IssueOfBook and Issue can be combined into the schema of Issue.

The Relationship set BookReturn is redundant. BookReturn is a One to many relationship between Book and Return and participation of Return is total. Therefore the Relation schema of BookReturn and Return can be combined into the schema of Return .

The Relationship set ReturnForIssue is redundant. ReturnForIssue is a One to one relationship between Issue and Return and participation of Return is total. Therefore the Relation schema of ReturnForIssue and Return can be combined into the schema of Return .

The Relationship set ReturnedBy is redundant. ReturnedBy is a One to many relationship between User and Return and participation of Return is total. Therefore the Relation schema of ReturnedBy and Return can be combined into the schema of Return .

Responsibility of G1 & G2: ER diagram



Contribution:

Team- G1: Visited Library and collected information about library management system by interviewing the library staff. Created design containing entities, relationships and attributes for library database system.

1) Sankarshan Kulkarni (20110184)
2) Riya Dhantoliya (20110168)
3) Gajanan Donge (20110061)
4) Jayesh Bhadange (20110082)
5) Siddharth Joshi (19110169)
6) Pintu kumar meena (19110193)

Team - G2: Created schema for relations and entities. Created ER diagram and relational tables. Defined key constraints for schema.

1) Jayesh Bhadange (20110082)
2) Pranav Rathod (20110143)
3) Naval Jaggi (20110118)
4) Harendra Khatik (20110072)
5) Sparsh Dawra (20110203)
6) Manpreet Singh (20110109)
7) Manish Jangir (20110107)