

(NPI000167),(NPI000168),
(NPI000170),(NPI000172),group
0assignment database
by Safal Acharya

Submission date: 18-Jun-2022 04:47PM (UTC+0800)

Submission ID: 1858351733

File name: 1st_sem_data_base_group_assignment.docx (283.7K)

Word count: 2867

Character count: 15536

ASSIGNMENT TEAM MEMBERS:-

- 1> SAFAL ACHARYA (NPI000170)
- 2> RIDEN PAUDEL (NPI000168)
- 3> RANGIN BASNET (NPI000167)
- 4> SAJAG SHRESTHA (NPI000172)

Abstract

The project aims to build a complete **library management database system** for UPA university as per their request. The project is named "*Cosmic library*". Before the project begins all the business rules of the university library are determined. Once the business rules are finalized, the entities, attributes and relationship between the entities are also finalized. Then ERD is formed accordingly. After that relational schema is generated with the help of which normalization is carried out. With the help of which we are able to build a perfect library management database system. We created all the tables using my SQL.

Acknowledgement

We are very thankful to all those who assisted us during the completion of this assignment. First of all, we'd love to express hearty gratefulness to our subject teacher, Er. Isha Baral. Her support and regular motivation helped us to accomplish our assignment before allocated time. Moreover, we'd like to acclaim our group members for sound cooperation and active participation throughout the process.

Drawbacks of using file base system

➤ Data Redundancy:

In the file system, the same data are repeated in multiple data files which causes redundancy of data. For example, a single student can borrow books multiple times on different dates. So here, students' details are stored in multiple files, which may result in data redundancy.

➤ Data Inconsistency:

During file processing system, different copies of similar data may contain several types of values which leads to data inconsistency. If a data needs to be modified then each file containing that certain data should also be modified which is a tedious task. This may result in errors i.e.; same data may have different values in different files.

For Example: If the Student Id is changed in library, then his/her id throughout all the files related to the student.

➤ Accessing Anomalies:

Accessing Anomalies refers to not having easy access to data in an accessible way. In file based system it is very tough to manage the data and to have control over them. If a user requires data in a particular way, then he/she needs to create a program for it.

➤ Poor Data Integrity:

In data base system we have data dictionary which stores data type of each stored data. Thus, there is no chance of inconsistency. E.g., Only phone number can be stored in the field where phone number is to be stored, no other data can be stored in that field. Whereas in file-based system data type are not defined. In DBMS security of data is maintained from the phase of development of database.

Advantages of database and DBMS

➤ **Hassel free data access:**

In DBMS you can access any data with a command of SQL and only the desired data is provided to the user. It is designed such that data are authorized according to the user. End user can have better experience because the data are more managed and organized.

➤ **Higher data security:**

In DBMS access to data to the end user is pre-determined. So, no unauthorized access of data is possible. Access of data is monitored and controlled in the building phase of data base therefore there is no issues like leak and mis use of data.

➤ **Limited redundancy of data:**

Occurrence of exactly same data in multiple location not only misuse the memory space in our database but also creates different types of errors such as insertion anomalies and so on. Data Normalization helps to remove such repetition.

➤ **Lowered Data Inconsistency:**

DBMS helps to minimize data inconsistency by reducing isolated files having repeated data. We can avoid inconsistency by using sufficient constraints in the database.

Business rules and Normalization

➤ Business Rules:

We can identify the entities, attributes, and relation with the help of business rules. Entity relationship diagrams are based on business rules. Based on the given scenario the business rules are listed below:

- The library won't lend some books, such as reference books, journals, etc. So, the students should be able to differentiate between the books which can be lent and which can't.
- Students are permitted to lend a maximum of five books at a time.
- The books are tagged with assorted colors and for Lecturers, they are permitted to borrow green tagged books. They can return the book after an interval of one month.
- Students are reminded of their return date through an e-mail a day before the due date.
- Students are charged fines if delayed in returning the books based on books and delayed time.
- Students have the access to reserve the loaned books. The staff will let the students know when the books become available for borrowing.

Normalization: The process where data in a database are all related data items are stored together and two basic requirements are met.

Un Normal form (UNF): The data model where the relational model does not meet any of the conditions of normalization is UNF.

Based on the case study let us take an example of Un Normal Form (UNF).

Student_ID	Name	Status	Address	Book	Publisher	Phone_number
SushilUPA220085	Sushil	BE	Kathmandu	quantum+	ABC	9856008722
BikashUPA220016	Bikash	MD	Paris	Nuclear	JPT	9865523471
IsaiahUPA220025	Isaiah	BE	Beijing	Business+	BMW	9875268210
SacarUPA220025	Sacar	PHD	Delhi	All in c	JMC	9988712956, 9866502144

(Table1: UNF)

Un Normal Form is illustrated in the above table having values in same row consisting of varying attributes. It can be further normalized as it results in issues like repetition of data.

First Normal Form(1NF)

Certain settings should be followed to perform the first normal form that are enlisted below.

- ✓ Each cell should be single valued i.e., atomic.
- ✓ Duplicate rows are avoided in the table.
- ✓ Arrays are not repeated.

Therefore, the 1NF table is created on the basis of above settings.

Student_ID	Name	Status	Address	Book	Publisher	Phone_num
SushilUPA220085	Sushil	BE	Kathmandu	quantum+	ABC	9856008722
BikashUPA220016	Bikash	MD	paris	Nuclear	JPT	9865523471
IsaiahUPA220025	Isaiah	BE	Beijing	Business+	BMW	9875268210
SacarUPA220025	Sacar	PHD	Delhi	All in c	JMC	9988712956
SacarUPA220025	Sacar	PHD	Dehli	All in c	JMC	9866502144

Table 2: Details(1NF)

The table is normalized into first form according to the requirement and each row has diverse values. There are 2/2 rows in above table with same student_id so the row was diversely identified.

Second Normal form(2NF)

After the normalization of first form, it should be normalized to second form. Certain settings should be followed to perform the second normal form that are enlisted below.

- ✓ It should be in first normal form.
- ✓ Partial dependencies should not exist between non-key attributes and key attributes.

Student_ID	Name	Status	Address	Book	Publisher	Phone_num
SushilUPA220085	Sushil	BE	Kharbari	quantum+	ABC	9856008722
BikashUPA220016	Bikash	MD	Thamel	Nuclear	JPT	9865523471
IsaiahUPA220025	Isaiah	BE	Baglung	Business+	BMW	9875268210
SacarUPA220025	Sacar	PHD	Ratnapark	All in c	JMC	9988712956
SacarUPA220025	Sacar	PHD	Ratnapark	All in c	JMC	9866502144



Conversion of 1NF to 2NF

Student_ID	Name	Status	Address
SushilUPA220085	Sushil	BE	Kharbari
BikashUPA220016	Bikash	MD	Thamel
IsaiahUPA220025	Isaiah	BE	Baglung
SacarUPA220025	Sacar	PHD	Ratnapark

Table 3.i: student details(2NF)

Book	Publisher	Phone_num
quantum+	ABC	9856008722
Nuclear	JPT	9865523471
Business+	BMW	9875268210
All in c	JMC	9988712956

Table 3.ii: Contact details(2NF)

Publisher	Phone_num
JMC	9988712956
JMC	9866502144

Table 3.iii

Third Normal Form(3NF)

After the normalization of second form, it should be normalized to third form. Certain settings should be followed to perform the third normal form that are enlisted below.

- ✓ It must be in 2NF.
- ✓ There must not exist any transitive dependencies between the non-key attributes and key attributes.

Student_ID	Name	Status	Address
SushilUPA220085	Sushil	BE	Kharbari
BikashUPA220016	Bikash	MD	Thamel
IsaiahUPA220025	Isaiah	BE	Baglung
SacarUPA220025	Sacar	PHD	Ratnapark

Table 3.i: student details(2NF)

The above table 3.i is already reduced to 3NF so it doesn't need to be further normalized.

Book	Publisher	Phone_num
quantum+	ABC	9856008722
Nuclear	JPT	9865523471
Business+	BMW	9875268210
All in c	JMC	9988712956



Conversion of 2NF to 3NF

Book	Publisher
quantum+	ABC
Nuclear	JPT
Business+	BMW
All in c	JMC

Table 4.i(3NF)

Publisher	Phone_num
ABC	9856008722
JPT	9865523471
BMW	9875268210
JMC	9988712956

Table 4.ii(3NF)

Entity Relation Diagram (ERD)

Entity relation diagram is the pictorial representation of the relation existing between real life entities. It helps to visualize the structure of our software application earlier to their actual development. It works as the model or blueprint for our database development.

Certain steps are mandatory for developing the best ERD (Entity Relationship Diagram).

They are as follows:

Step 1: Determine the entities.

Step 2: Determine the attributes of the entities.

Step 3: Determine the relationship between the entities.

Finalizing the entities, attributes and their corresponding relations helps to develop the [ERD](#) with ease.

Identification of entities

Entities:

Entities differ with the project you work on. Rectangular shape box is used in [ERD](#) to denote entities.

For the provided case study, we found the following entities relevant:

ENTITIES

- a. Book
- b. Publisher
- c. Member
- d. Classification
- e. Transaction
- f. Author
- g. Reservation

Description of each entity:

Entities	Description
Book	Information of each book available in the university library is stored with unique book id with other details.
Publisher	Details of publishers of each book in library is assigned with unique publisher id with other details with which book can be searched.
Member	This entity holds all the details of the individual who borrows books from the library.
Reservation	It holds the record of book reserved by any library member.
Classification	There are distinct categories of members in the library i.e., student, lecturer, and staff. It holds their identity.
Transaction	It holds all the transaction records of book borrowing done in the library.
Author details	Details relating to the author of the respective books are stored in this entity.

Identification of attributes for each entity:

Attributes are the properties which describe an entity. Each of the above entity is characterized by certain attributes. These attributes should be simple. The attributes can be single valued, multivalued or can be null sometimes.

The attributes of the entity set listed above are as follows:

Primary key is shown in bold letters with underline whereas Foreign Key are marked with (*) symbol.

ATTRIBUTES:

1. Book:

- a) **Book_ID**
- b) ISBN
- c) Field
- d) Title
- e) Categories
- f) Publisher_ID*
- g) Author_ID*
- h) Edition
- i) Status

2. Publisher:

- a) **Publisher_ID**
- b) Publisher_Name
- c) Publisher_Address
- d) E-mail_ID
- e) Contact_num

1
3. Transaction:

- a) **Transaction ID**
- b) **Member_ID**
- c) **Book_ID***
- d) **Borrow Date**
- e) **Return Date**
- f) **Actual Return Date**
- g) **Fine**

4. Author Details

- a) **Author ID**
- b) **Author Name**
- c) **Address**
- d) **Contact no.**

5. Reservation:

- a) **Book_ID***
- b) **Reservation Date**
- c) **Member_ID***

6. Member

- a) **Member ID**
- b) **E-Mail_ID**
- c) **Classification_ID***
- d) **Name**
- e) **Department**
- f) **Status**

7. Classification

- a) Classification_ID
- b) Classification Name

Description of attributes of each entity

Entity: Book :

Attributes	Description
Book_ID	Book ID is a unique identity assigned to every book available in the library. It is also the primary key on this table.
ISBN	ISBN is the unique integer value assigned for each book. Since there are multiple copies of the same book in the library having the same (ISBN) we still require another unique identity for them which is provided by the assigned book ID.
Title	It holds the name of book.
Field	It holds the field i.e., IT, Physics, Math's etc.
Publisher_ID	It is a foreign key which connects the Publisher table with the Book table that displays the publisher details.
Edition	It keeps the record of the year of edition of the book.

Author_ID	It is a foreign key which connects the Author table with the Book table that displays the author's details.
Category	Books are categorized as open stack, red tagged, green tagged, yellow tagged, journal, research paper, maps etc. They have different properties i.e.; some are lendable while some are not. They have different lending periods.
Status	It stores books status as available, not available, or not lendable for borrowers.

Entity: Publisher Detail:

Attributes	Description
Publisher_ID	It is the primary key for the table. Since more than one publisher can have same or similar names It helps to uniquely represent the publishers.
Name	It holds the name of the publisher.
Address	It holds the publisher's address.
E-mail_ID	It holds the mailing address of the publisher.
Contact no.	It holds publishers contact number.

Entity:Transaction:

Attributes	Description
<u>Transaction_ID</u>	It is the primary key on the table. It helps to uniquely identify each transaction done in the university library.
<u>Member_ID</u>	It is the foreign key in the table. A member can do multiple transactions using his/her member ID.
Book_ID	It holds the record of borrowed book.
Borrow Date	It stores the date in which the book was borrowed.
Return Date	The date on which lenders must return the book is stored. Return date may vary according to the category of the books.
Actual Return Date	Sometimes Member may return the book earlier or sometimes prior to the return date, so it holds the actual date of return. Fines are charged when books are returned prior to the return date.
Fine	If books are returned after the due date, a certain fine is charged.

Entity: Member:

Attributes	Description
Membner_ID	It is the primary key for the table. It lets members do library transactions uniquely.
E-mail_ID	It holds members E-mail Id
Classification	It holds classes of library members i.e.,
Name	It records the member's name.
Department	It holds the department for which the members belong.
Status	It shows whether the members membership is active or expired.

Entity: Classification:

Attributes	Description
Classification_ID	It gives a unique ID to each classification of library member.
Classification Name	It holds the class of library members as student, lecture or staff.

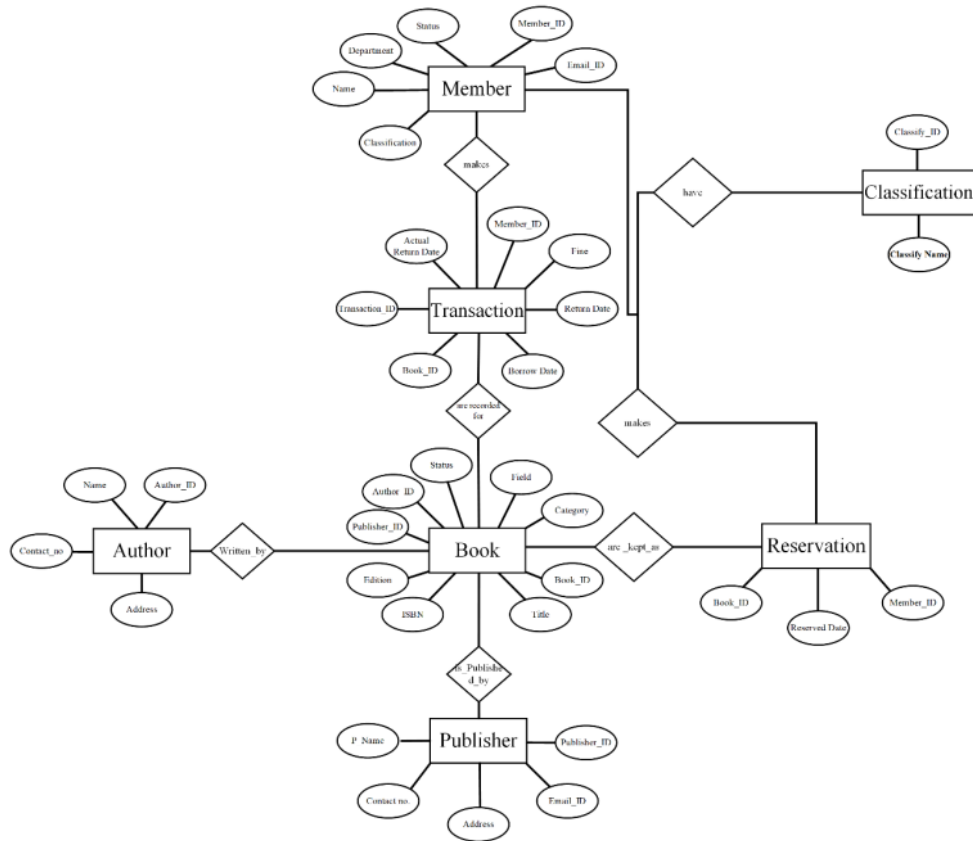
Entity: Author Details:

Attributes	Description
Author_ID	It is the primary key for the author table. It helps uniquely identify the author.
Name	It holds the author's name.
Address	It records author's address.
Contact no.	It holds the author's contact details.

Entity:Reservation:

Attributes	Description
Member_ID	It holds ID of the member who reserves the book.
Book_ID	It holds the book id which is reserved by the member.
Reservation_Date	It holds the date on which the book is reserved.

Simple ERD (Entity Relationship Diagram)

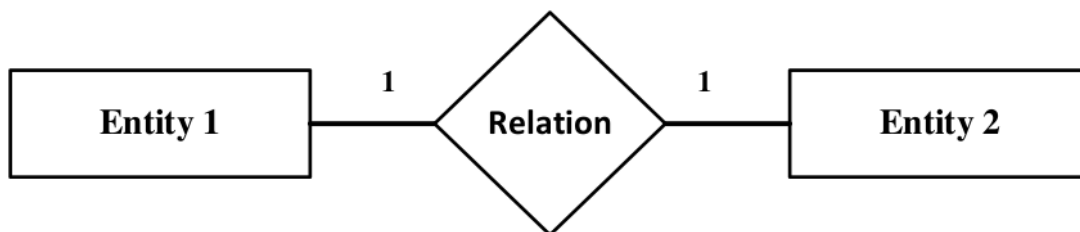


Documentation of bond amongst entity

The kins among the two different entities should be discovered to draw an entity relation diagram as it helps giving structure to the entity diagram. The four main types of relations that link the entities together are described below.

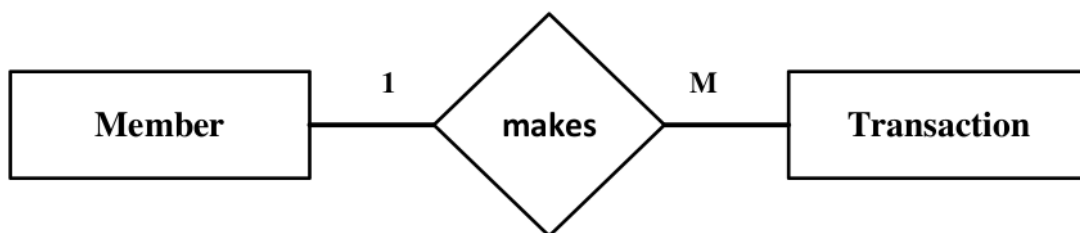
One to One Relation (1:1)

In such types of relation, one record of an entity table is linked with one and only one record of another entity table.



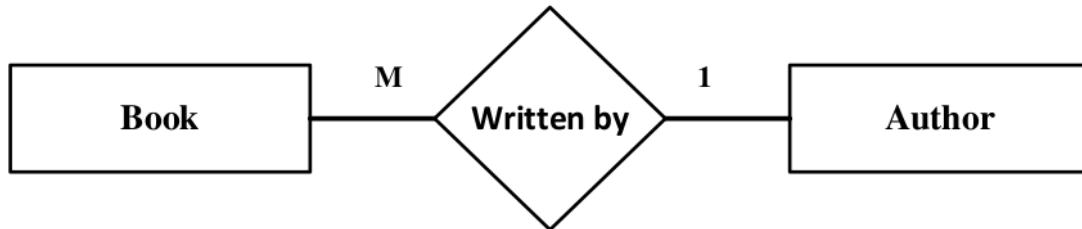
One to Many Relation(1:M)

In such types of relation, one record of an entity table is linked with more than one record of another entity table.



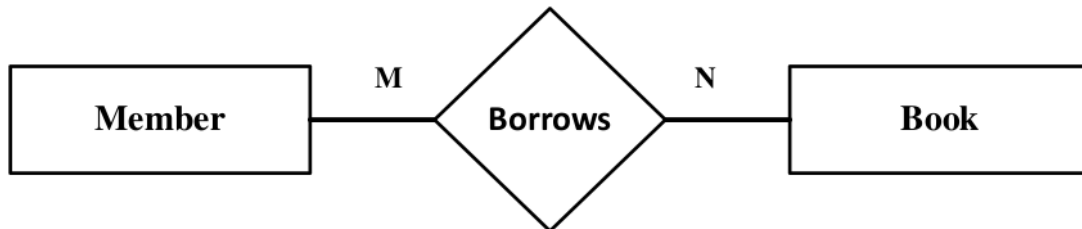
Many to One Relation(M:1)

In such types of relation, many records of an entity table are associated with only one record of another entity table.



Many to Many Relation(N:M)

In such types of relation, many record of an entity table are associated with more than one record of another entity table.



The relation amongst different entities based on the certain settings are shown below:

Relationship	Description
Book M:1 Author	Many books are written by one author.
Book M:1 Publisher	Many books are published by one publisher.
Transaction M:N Book	Many transaction are recorded for many books or many books can be recorded for many transactions.
Book M:N Reservation	One book is kept as many reservations or many reservations are arranged for many books.

Constraints

Logical Constraints:

- The library doesn't let the members lend books such as reference books, student projects, maps and etc.
- Books are labeled with different colored tags to differentiate between them.
- Members are restricted to borrow 5 or 10 books for Students and Lecturers respectively.
- Members have access to reserve the borrowed books.

Null Constraints:

All the values in an attribute cannot be null because the data they hold is necessary for the entity with exceptions as such:

- The Fine attribute in the Member table of the Cosmic_Library database can contain Null value in case the member returns the book at appropriate time.
- The Extra _days attribute in the Member table of the Cosmic_Library database can have Null value if the member returns the book with no due time.

Unique Constraints:

- The Member_ID should be distinct.
- The Book_ID should be distinct.
- Each Author's ID should be distinct.
- Each Publisher's ID must be distinct.

Primary Keys Constraints:

- All primary keys are varchar datatype.
- Considering all the primary keys are of varchar datatype, they can be uniquely distinctive to each other.

References

Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom: 2009,
Pearson Prentice Hall *Database System the Complete Book*. Pearson Prentice Hall: Available at:
<https://people.inf.elte.hu/miiqaai/elektroModulatorDva.pdf>

Ben Brumm, Jun 23, 2021, Database Design for a Library Management System [Video]
<https://youtu.be/yldrIdoXiYk?t=26>

Paul Beynon-Davies, Third Edition, Database Systems.

Raghu Ramakrishanan & Johannes Gehrke. Database Management System, Second Edition.

Sikkim Manipal University, Database Management System, Edition 2009

Toby Teorey, Sam Lightstone and Tom Nadeau, Database Modeling & Design, Fourth Edition

M.V.Kamal, Department of Computer Science & Engineering
Malla Reddy College of Engineering & Technology. [pdf]
https://mrcet.com/downloads/digital_notes/ECE/III%20Year/DATABASE%20MANAGEMENT%20SYSTEMS.pdf

SQL Tutorial,
<https://www.w3schools.com/sql/>

Deborah Morley, Charles S. Parker, 6/10/2015[pdf]:
https://www.oakton.edu/user/2/rjtaylor/CIS101/Text/PowerPointPresentations/morley15e__ppt_ch14%20REV.pdf

Learn S.Q.L
<https://www.tutorialspoint.com/sql/index.htm>

(NPI000167),(NPI000168),(NPI000170),(NPI000172),group
0assignment database

ORIGINALITY REPORT

1 %	0 %	0 %	1 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Singapore School Manila Student Paper	<1 %
2	Submitted to St. Patrick's College Student Paper	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		