Library Management System

A mini-project report submitted for

Database Management System (Semester V)

by

Chyankk Kumar (Roll No. 8283)
Aarti Bandekar (Roll No. 8296)
(Roll No.)

(Group Code: 02)

Under the guidance of
Prof. Vaibhav Godbole

(Sign with date)

DEPARTMENT OF ELECTRONICS ENGINEERING
Fr. Conceicao Rodrigues College of Engineering
Bandra (W), Mumbai - 400050
University of Mumbai
October 3, 2019

Library Management System

Chyankk Kumar[1], Aarti Bandekar [2]

Abstract
This database revolves around a Library Management System ,we also study how to connect this to our java frontend using a JDBC Driver

Keywords
Library Management System

[1] *Fr.Conceicao Rodrigues College of Engineering*
[2] *Department of Electronics Engineering*
[3] *Mumbai, Maharashtra*

Contents

1. Problem Statement

Create an ER diagram for a database with following specifications. The database stores name, age, gender and address of each person. Each person is identified by his/her name. A person may be either a reader or a writer (i.e., we assume that a writer is not stored as a reader in the database and a reader is not stored as a writer in the database). The database also stores books and, for each book, we store its name, publisher and its binding type. Each book may have several editions and for each edition we need to record the number of pages and price. Each book is written by only one writer (author). An author may have written one or more books. For each author, we also store his place of birth and the name of his/her best-selling book (each author has exactly one bestselling book). A reader may read several books. For each book a reader reads, he/she gives a rating for that book and the database stores these ratings for each book. For each reader, we store his/her favourite writers. Each reader may have zero or more favourite writers. If you make some assumptions, list them clearly. Check whether the tables are in 3NF. If not then design them in 3NF.

Draw dependency diagram in 1NF, 2NF and 3NF form. Create relevant tables and insert appropriate data in them. (Minimum 100 rows in each table should be added).

1. Find number of pages of a book called ”Database management systems”
2. Find names of writers whose books are best sellers
3. Find names of books written by female writers.
4. Find address of the reader who has given ratings more than 4
5. Find names of readers who are reading edition of ”Database management system” with edition number ”E10001”.
6. Find binding types of bestseller books read by only male readers
7. Find address of those authors who has written at least one book for TMH publication

2. 1st Normal form(1NF)

2.1 Explanation

Note that the data in Figure 1 reflects the data viewed in the 1st Normal form. We start by first making a ”Person” entity that stores attributes like the ”ID”, that is used to uniquely identify each person.The problem statement also required us to store this unique person with his/her name that has been broken down to the first and last name respectively.The next attribute ”Age is used to denote the age of this person ”, ”Gender” is used to identify his or her gender as either male or female in the database, ”Address” stores this persons residential address in our database and finally,”Type” helps us identify whether this given person is a Reader or an Author as our database requires us to store them as either one, the reader cannot be an

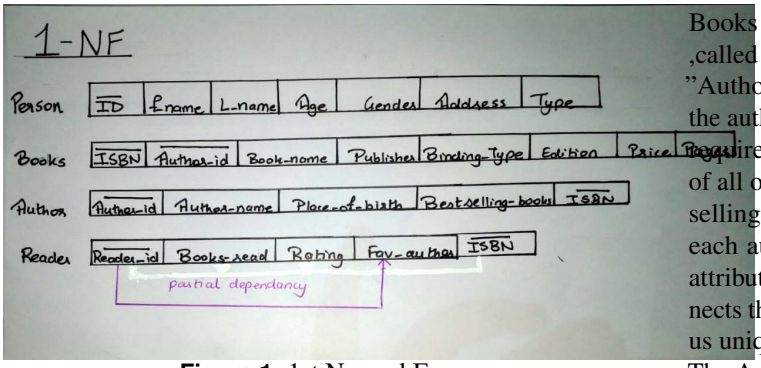

Figure 1. 1st Normal Form

author and vice versa. attribute ”ID” is unique to an individual and contains no null values hence it is chosen as our primary key for this entity. Therefore, knowing the ID allows you to uniquely identify any person. It also gives a count of the total number of users registered for our given library in the database.

The entity ”Books” contains its own set of attributes, this table helps us organize and store all the books uniquely in our database.However, it must be noted that a single copy of each book has been assumed eliminating the possibilities of owning multiple copies and conflicting with the uniqueness of the ”ISBN” code. ”ISBN” code stands for International Standard Book Number (ISBN) it is a unique numeric book identifier bar code used by publishers, booksellers, and libraries for book ordering and stock control. ISBN is a worldwide identification code assigned to books, any monographic publication in printed, non-printed and mixed media formats. ”Author-id” is a foreign key from the person attribute and references to ”ID” present there. This helps establish a link in our database wherein we’re only extracting information of those users marked as an author, this attribute allows us to uniquely link each book to their respective authors through their unique identities. As the name suggests, the attribute ”Book-name”, stores the identifying name of the book referred by the ISBN code for user convenience. ”Publisher”, is another attribute that stores the publisher’s name while ”Binding-type” helps us identify what binding that particular book contains, similarly ”Edition” allows us to input the edition of the book ,”Price” and ”Pages” allows us to look up the Maximum Retail Price while pages stores the total number of pages of the given book respectively.
The ISBN allows us to uniquely identify all the books as the value is different for each Edition and Binding type, hence eliminating the need to break our table into multiple tables while always keeping it unique hence it has been chosen as our primary key,as the author-id is also unique to each other it also forms our primary key, hence forming a composite primary key.Please note here we are assuming that each author has written exactly one book hence keeping the attribute unique.
The entity Author,stores data about our author, just like the Books entity is draws a foreign key from The person entity ,called ID”, which as discussed earlier keeps the data unique , ”Author -name” is used for the user convenience to identify the author with that specific ID.”Place of Birth”,is an attribute requires by our problem statement to store the place of birth of all our authors.”Bestselling-books” stores exactly one best-selling book of that particular author,here it is assumed that each author has exactly one bestselling book.”ISBN” is an attribute which is a foreign key from the book table that connects the Books attribute to the Author attribute and also helps us uniquely identify each book specific author.
The Author-id as well as the ISBN are unique,non null values and are hence chosen as our primary keys, which together forms our composite primary key.
The entity Reader,stores data about our reader, it draws a foreign key from The person entity ,called ”ID”, which as discussed earlier keeps the data unique, This helps establish a link in our database wherein we’re only extracting information of only those users marked as a Reader under attribute ”Type”, this attribute allows us to uniquely link each Reader through their unique identities. Books-read allows us to make a note of all the books read by our reader and it is a foreign key referencing to ”Book-name” attribute under the ”Books” entity.. Our database also allows us to rate a book under the attribute ”Rating” where a check constraint makes sure that no value below 1 and above 5 is inputted. Our user can also store the name of their favourite authors under the attribute ”Fav-author” that is a foreign key from the author table by attribute ”Author-name”.This value is not unique as many readers can have the same favourite author .The ”ISBN” references to ISBN from the Books entity ,thereby forming a link.
The Reader-id as well as the ISBN are unique,non null values and are hence chosen as our primary keys, which together forms our composite primary key.
The table entries invite data inconsistencies. For example,with the ID’s used everywhere however linking the foreign keys eliminate that issue.
The table displays data redundancies. Those data redundancies yield the following anomalies:
Update anomalies and Insertion anomalies.
Ex: Author-name,which makes it difficult to track down the actual names and is hence further divided into firstname and lastname same applies to fav-author in the Reader entity.
In spite of those structural deficiencies, the table structure appears to work; the report is generated with ease. Unfortunately, the report might yield varying results depending on what data anomaly has occurred. Such reporting anomalies cause a multitude of problems for librarians.
Only one person identified by the ID, that person’s characteristics from the Person entity should not have to be typed in each time the main file is updated hence using only the ID’s everywhere eliminates that issue .
The data redundancy leads to wasted disk space and 1NF helps us escape that problem. The possibility of introducing data integrity problems caused by data redundancy must therefore

always be considered when a database is designed.
Normalizing the table structure to 1NF reduces the data redundancies. If repeating groups did exist, they were eliminated by making sure that each row defines a single entity.The dependencies are also identified to diagnose the normal form.
The process is summarized below:

1. Eliminate the Repeating Groups The data is represented in a tabular format, where each cell has a single value and there are no repeating groups. To eliminate the repeating groups, the nulls are eliminated by making sure that each repeating group attribute contains an appropriate data value.

2. Identification of the Primary Key ID is the primary key for the person entity
   ISBN and Author-id is the primary key for entity author.
   Reader-id and ISBN is the primary key for entiry reader.

3. Identify All Dependencies In our database only one dependency was observed in the reader table where there is a partial dependency between the reader-id and fav-author. In other words, the reader-id is dependent on the fav-author. You can write that dependency as: Reader-id -¿fav-author.

3. 2nd Normal Form

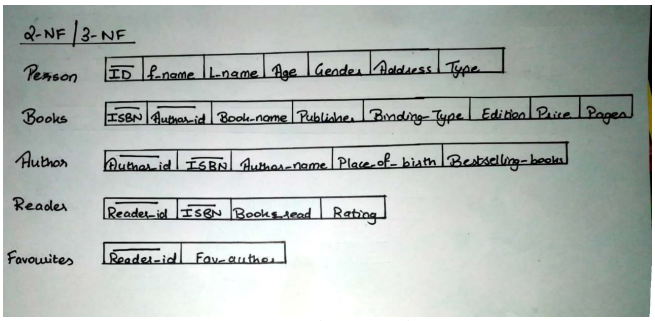

Figure 2. 2nd Normal Form

3.1 Explanation

Converting to 2NF is done only when the 1NF has a composite primary key. If the 1NF has a single-attribute primary key, then the table is automatically in 2NF.The following steps were involved:

1. Make New Tables to Eliminate Partial Dependencies
   For each component of the primary key that acts as a determinant in a partial dependency, create a new table with a copy of that component as the primary key. While these components are placed in the new tables, it is important that they also remain in the original table as well. It is important that the determinants remain in the original table because they will be the foreign keys for the relationships that are needed to relate these new tables to the original table.In this table only one partial dependency was observed in the reader entity between reader-id and fav-author. Each component thus becomes the key in a new table. In other words, the original table is now divided into a new table named Favourites.

2. Reassign Corresponding Dependent Attributes
   As the dependencies for the original key components are found by examining the arrows below the dependency diagram shown in Figure 1, The attributes that are dependent in a partial dependency are removed from the original table and placed in the new table with its determinant. Any attributes that are not dependent in a partial dependency remain in the original table.
   Favourites (Reader-id,Fav-author)

This table is in second normal form (2NF) as:

1. It is in 1NF.

2. There are no partial dependencies anymore; that is, no attribute is dependent on only a portion of the primary key.

4. 3rd Normal Form

4.1 Explanation

The 3NF format eliminates all the transitive dependencies i.e the dependency that exists between two non-prime attributes,however,Figure 2 shows no transitive dependency, which could generate anomalies. The following steps are followed to get the tables in a 3NF format:

1. Make New Tables to Eliminate Transitive Dependencies:
   For every transitive dependency, write a copy of its determinant as a primary key for a new table. A determinant is any attribute whose value determines other values within a row. If you have three different transitive dependencies, you will have three different determinants. As with the conversion to 2NF, it is important that the determinant remain in the original table to serve as a foreign key

2. Reassign Corresponding Dependent Attributes
   Identify the attributes that are dependent on each determinant identified in Step 1. Place the dependent attributes in the new tables with their determinants and remove them from their original tables.However,there are no transitive dependencies identified here.

A table is in third normal form (3NF) when:

1. is in 2NF.

2. It contains no transitive dependencies

5. Entity Relationship Diagram

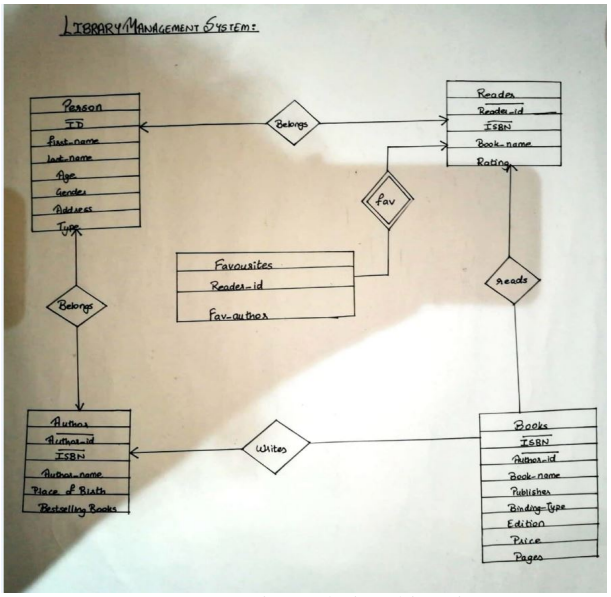

Figure 3. Entity Relationship Diagram

The ER or (Entity Relational Model) is a high-level conceptual data model diagram. Entity-Relation model is based on the notion of real-world entities and the relationship between them.
ER modeling helps you to analyze data requirements systematically to produce a well-designed database. ER model allows you to draw Database design. It also helps you identify the entities which exist in a system and the relationship between them.
An ER diagram is based on 3 concepts:

1. Entities: Recognizable dataset to store data vin the database
2. Attributes: The Single valued property of either an entity or a relation type
3. Relationships: Association among two or more entities

An important feature of ER Diagram is the Cardinality,it defines the numerical attributes of the relationship between two entities or entity sets.

Different types of cardinal relationships are:

1. One-to-One Relationships
2. One-to-Many Relationships
3. May to One Relationships
4. Many-to-Many Relationships

In the ER Diagram, One-to-Many cardinal relationships is used to visualise the cardinality of the relationship of an entity with the entity Person. The Person entity is at a point of unique significance which holds the information needed by other tables such as, id, first-name, last-name, address, age etc. For each reader, we store their favourite Author.
Reader has zero or more favourite writers (one to many cardinality). The Favourite entity draws data from the Reader entity.
Each book is written by only one writer (one to one cardinality) The books entity draws id from Person entity. to ensure that no two people have repeated and one book is written by one author.
An author may have written one or more books.(many to one cardinality) The Author entity draws data from the entity person and the entity books.

Here, In the final ER- Diagram there are 5 entities namely: Person, Author, Reader, Books and Favourite.
Person has ID, FIRST-NAME, LAST-NAME, AGE, GENDER, ADDRESS, TYPE. ID is a primary key or the table Person.
Reader has READER-ID, ISBN, BOOK-NAME, RATING.
Books has ISBN, AUTHOR-ID, BOOK-NAME, PUBLISHER, BINDING TYPE, EDITION, PRICE, PAGES
Author has AUTHOR-ID, ISBN, AUTHOR-NAME, PLACE-OF-BIRTH, BESTSELLING-BOOKS

ID is a primary key of Person.
ISBN and AUTHOR-ID are a foreign key of Author.
ISBN and READER-ID are a foreign key of Reader.
ISBN is a primary key of Books. AUTHOR-ID is a foreign key of Books.

Author entity draws a few functional data from Person, and hence is Related to Person .
Whereas Books draws data from Author as well as Reader hence is Related to Author and Reader.
Reader draws data from Person as well. Therefore, Person and Reader are Related.
The Entity Favourites is Weakly related to the Reader entity.

An entity that cannot be uniquely identified by its own attributes and relies on the relationship with other entity is called weak entity. The weak entity is represented by a double rectangle.
The entity Favourite is a weak entity as it solely relies on the Reader entity to draw information from other entities that are, Person and books. As there is no direct relationship to the information needed, it forms a weak relational entity.
The entity Favorite draws the READER - ID from Reader entity which in turn draws the data from the Person table. It also draws the AUTHOR-NAME with Reference to the Reader entity which draws its data from Person entity.

A weak entity does not save enough attributes to have its own primary key. The primary key of a weak entity set is the combination of the partial key and the primary key of a string entity set. The relationship between weak entity and strong entity set is called as Identifying Relationship.

6. Queries

Query 1: Find number of pages of a book called ”Database management systems.

SELECT first-name, last-name, Bestselling
FROM author;

	first_name character varying (200)	last_name character varying (200)	bestselling character varying (200)
1	Lisette	Strongman	Year of the Wolf, The (Sude...
2	Hall	De Andreis	Greatest Movie Ever Sold, T...
3	Anatol	Bail	Hush!
4	Devin	Braidman	Boys of Baraka, The
5	Latia	Capnor	Paid in Full
6	Aldo	Woolens	Lola Versus
7	Pepe	Codling	Over Your Cities Grass Will ...
8	Jakie	McCloid	Albino Alligator
9	Alphard	Hasselby	Wilby Wonderful
10	Patten	Newart	Shall We Dance
11	Marjy	Harburtson	White Nights
12	Parry	Spraggs	Score: A Hockey Musical
13	Brnaby	Bartomeu	Last Lions, The
14	Gardie	Northern	Counter Investigation (Cont...
15	Taddeo	Porcas	Best of the Best 3: No Turni...

Figure 4. Output for Query 1

Query 2: Find names of writers whose books are best sellers

SELECT p.first-name, p.last-name, b.book-name
FROM person p , books b
WHERE p.id= b.author-id AND p.gender=’Female’;

Query 3: Find names of books written by female writers.

Insert into Books (isbn, author-id, book-name, publisher, binding, edition, price, Pages) values (’681890290-1’, ’59-9974514’,
’Database Management System’, ’Universidad ICESI’, ’soft cover binding’, ’E10001’, ’dollar 30.83’, ’260’);

SELECT Pages
FROM Books
WHERE book-name=’Database Management System’;

Query 4: Find address of the reader who has given ratings more than 4.

	first_name character varying (50)	last_name character varying (50)	book_name character varying (100)
1	Lisette	Strongman	Year of the Wolf, The (Sude...
2	Devin	Braidman	Boys of Baraka, The
3	Latia	Capnor	Paid in Full
4	Marjy	Harburtson	White Nights
5	Laryssa	Curtiss	Skipped Parts
6	Beryle	Prestland	Drifting Clouds (Kauas pilve...
7	Lita	Haglington	Agora
8	Betty	Gallant	Other Dream Team, The
9	Cynthie	Hastelow	Fantomas (Fantômas - À l'o...
10	Glenn	Rofe	Competition, The
11	Lynnell	Allan	Nanking
12	Margot	Moir	I Am Divine
13	Thia	Forster	I Always Wanted to Be a Ga...
14	Agata	Attfield	Hollow Point
15	Charmion	Debell	Buried Alive

Figure 5. Output for Query 2

	pages character varying (100)
1	260
2	654

Figure 6. Output for Query 3

SELECT p.first-name, p.last-name, p.address
FROM person p, Reader r
WHERE p.id= r.reader-id AND r.rating ¿ 4;

	first_name character varying (50)	last_name character varying (50)	address character varying (50)
1	Jolynn	Swindin	632 South Center
2	Jolie	Ateggart	59 Farragut Park
3	Chiquita	Yoslowitz	4795 Ryan Center
4	Alasdair	Songhurst	1217 Express Street
5	Wiley	Meijer	2 Northwestern Trail
6	Ellswerth	Caws	98 Delaware Way
7	Kelly	Pennaman	6 Schiller Pass
8	Tami	Arden	10622 Basil Way
9	Blondell	Kinnett	7 Ridgeview Street
10	Gloriane	Halworth	9 Dovetail Hill
11	Clemens	Balasin	20162 Steensland Plaza
12	Kitty	Fussell	2614 Farwell Avenue
13	Albina	Myrick	352 Meadow Ridge Point
14	Morgen	Bungey	798 Emmet Road
15	Bathsheba	Saltmarshe	83917 South Avenue

Figure 7. Output for Query 4

Query 5: Find names of readers who are reading edition of ”Database management system” with edition number ”E10001”.

```
SELECT p.first-name , p.last-name
FROM person p, reader r, books b
WHERE r.reader-id=p.id AND r.isbn=b.isbn And b.book-name='Database Management System';
```

	first_name character varying (50)	last_name character varying (50)
1	Julia	Ogilvie
2	Bette	Borrington
3	Morgan	Borg-Bartolo
4	Duffy	Qualtrough
5	Barbra	O' Mahony

Figure 8. Output for Query 5

Query 6: Find binding types of bestseller books read by only male readers.

```
SELECT b.binding
FROM person p , books b
WHERE p.id=b.author-id AND gender='Male';
```

	binding character varying (100)
1	Spiral binding
2	soft cover binding
3	Case binding
4	Spiral binding
5	Spiral binding
6	Saddle stitch binding
7	Saddle stitch binding
8	Case binding
9	Spiral binding
10	Case binding
11	Spiral binding
12	soft cover binding
13	Saddle stitch binding
14	Spiral binding
15	Case binding

Figure 9. Output for Query 6

Query 7: Find address of those authors who has written at least one book for TMH publication.

```
SELECT p.address
FROM books b, person p
WHERE b.author-id=p.id AND publisher='TMH Publication';
```

	address character varying (50)
1	060 Onsgard Alley
2	9 Pierstorff Trail
3	494 Kim Place
4	851 Nova Crossing
5	327 Stone Corner Junction
6	98 Sullivan Alley
7	6975 Bashford Crossing
8	2 Carey Center
9	22757 Emmet Point

Figure 10. Output for Query 7

7. Constraints Applied

The CHECK constraint controls the value of a columns being inserted. It provides the CHECK constraint, which allows the user to define a condition, that a value entered into a table, has to satisfy before it can be accepted. The CHECK constraint consists of the keyword CHECK followed by parenthesized conditions.
The attempt will be rejected when update or insert column values that will make the condition false.
The CHECK constraint can be defined with a separate name. Our database applies only one constraint in the Reader entity on the rating attribute.The referred column can not contain any value less than or equal to zero(0) ,if the CHECK constraint is violated ,it throws an error for the same.
Given below is the constraint used :

```
1    Reader_id VARCHAR(100) primary key,
2    isbn VARCHAR(100),
3    book_name VARCHAR(100),
4    Rating numeric(2) check (Rating>0),
5    foreign key(Reader_id) references Person(id));
```

Figure 11. CHECK constraint

8. Applications

Book recommendation system : By adding genres and making it more user specific, books can be referred as per the previous choice and the best selling category.

9. References

1. https://www.lucidchart.com/pages/ER-diagram-symbols-and-meaning
2. https://beginnersbook.com/2015/04/e-r-model-in-dbms/
3. https://www.techopedia.com/definition/24361/database-management-systems-dbms
4. https://youtu.be/ewuSDIKuq4Y
5. https://www.cybertec-postgresql.com/en/triggers-to-enforce-constraints/
6. https://youtu.be/qw–VYLpxG4

10.