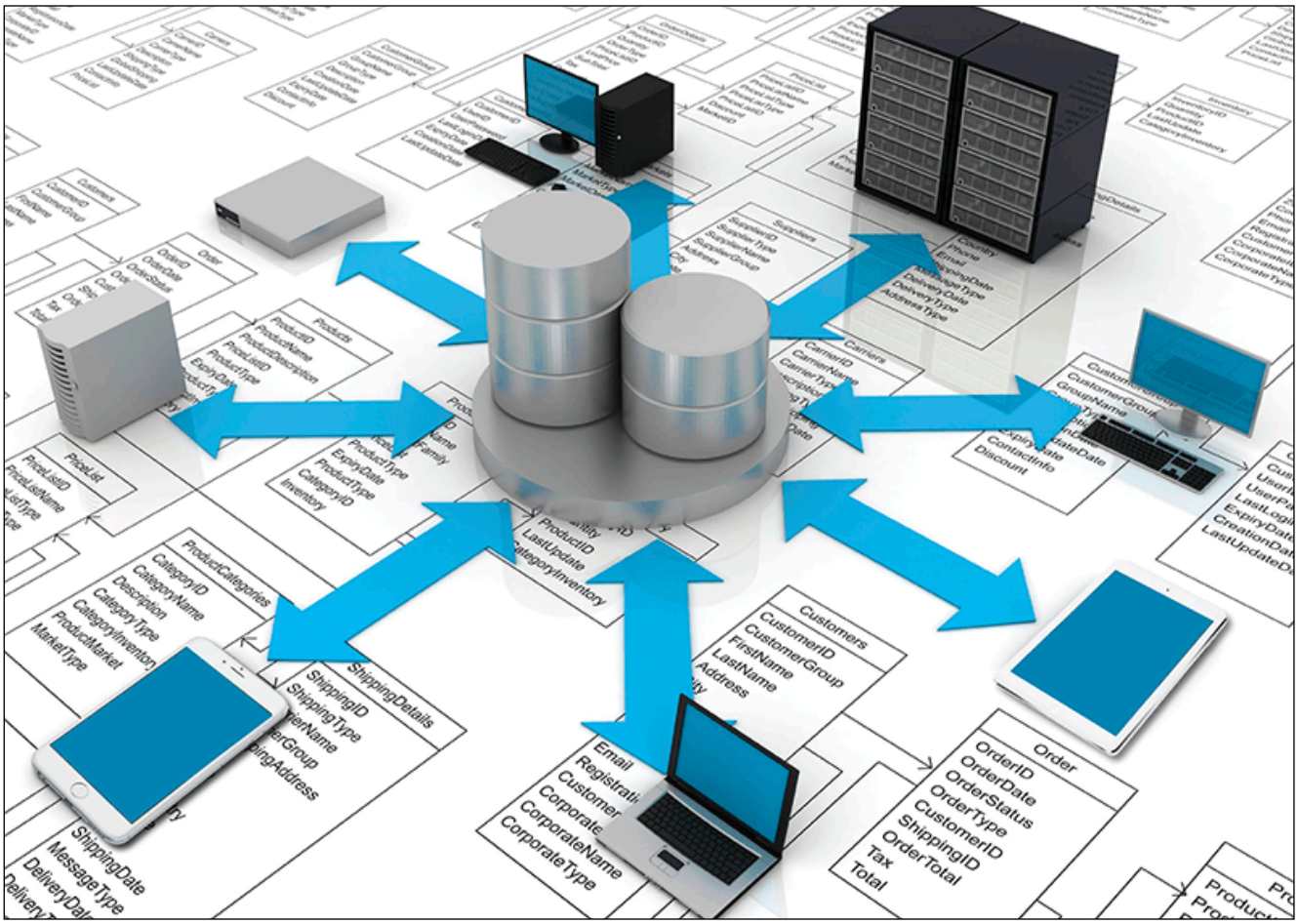
Assignment 4

Rohit Ravishankar - March 23, 2018



Index

Introduction

A database is a collection of data that is saved and organized to allow easy retrieval when needed. It is the collection of schemas, tables, queries, reports, views, and other objects.

A DBMS or Database Management System makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or programs, ensuring that data is consistently organized and remains easily accessible.This kind of system manages and protects data so that the database is safe and secure.

Databases consists of tables that include groups of related data fields that are known as records. Databases are not limited to only computers; in fact, a phone book is an example of a database. All of the names alphabetized and each column has its own category. There is a column for your name, phone number, and possibly a street address. With a relational database, all of the data within the row can be pulled up when you are looking for the specific attribute.

Some of the different kinds of databases management systems include:-

- Relational DBMS
- NoSQL DBMS
- Object DBMS
- XML DBMS

The main components of SQL include:-

- Data Manipulation Language (DML)
- Data Definition Language (DDL)
- Data Control Language (DCL)

Advantages of Databases

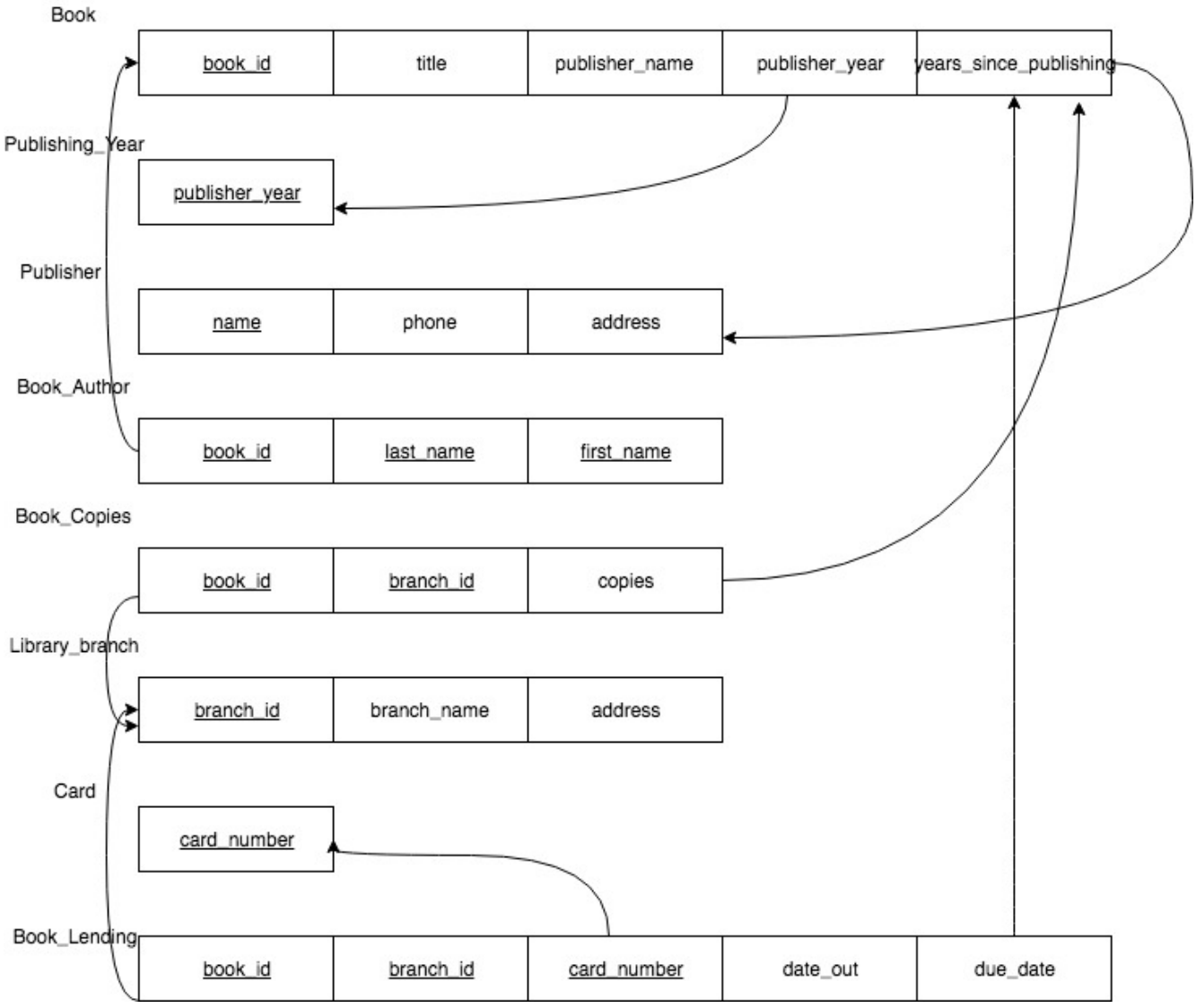
1. Ability to control redundancy
2. The integrity of the information being stored can be maintained
3. It can restrict access
4. It can share data
5. Can backup/recover information

Introducing the database

The database I have created here is a Library Database. The entities include:-

- Book - to store information about each book
- Publisher - to store information about each publisher
- Book_Author - to store information about each book author
- Publishing_Year - to store information about the different years of publications
- Library_Branch - to store information about different branches of a library
- Book_Lending - to store information about the books that have been lent from the library
- Book_Copies - to store information about the number of copies of a book in each library

Schema Diagram



The aforementioned figure, is a schema diagram for the implemented database. A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. The database is implemented after normalizing to protect data integrity and reduce redundancy. The primary keys are represented with underlined attribute names and the arrows represent foreign key relationships to the tables where the arrows point.

Normalizing the tables

Database normalization, or simply normalization, is the process of restructuring a relational database in accordance with a series of normal forms in order to reduce data redundancy and improve data integrity. Through the below steps we observe how data which is not normalized is normalized to satisfy the normal forms. Listed below are the 2 initial tables that are not normalized.

title	publishing_ year	years_ since_ publishing	publisher_ name	phone	address	author_ name (first, last)
12	2017	1	HACHETTE - LIVRE	585625400	Chicago	Edward, Angel
Databases	2017	1	MCGRAW-HILL	585625700	Rochester	John, Smith
Computer Networks	2017	1	PEARSON	585626800	Henrietta	Andrew, Garfield
Computer Graphics	2013	5	GRUPO PLANETA	585625300	Princeton	Edward, Petersman

branch_name	address	Title	copies	date_out	due_date
Henrietta	Rochester	Databases	2	12-JAN-2017	18-JAN-2017
Pittsford	Rochester	Computer Networks	2	02-FEB-2018	15-FEB-2018
Pittsford	Rochester	Computer Graphics	1	01-MAR-2018	05-MAR-2018
Brighton	Rochester	Databases	3	04-JAN-2017	06-FEB-2017
Fairport	Rochester	Computer Networks	4	07-FEB-2017	17-FEB-2017

Applying 1 Normal Form:-

- Each cell to be single valued - By splitting author_name into first and last name we satisfy the condition of each cell to be single valued.

- Entries in one column are of the same type - title should always be stored as characters not as integer. Eg 12 is stored as '12'

- Rows are uniquely identified - By introducing book_id, branch_id the rows are uniquely identified

branch_id	title	publishing_year	years_since_publishing	publisher_name	phone	address	first_name	last_name
1	12	2017	1	HACHETTE - LIVRE	585625400	San Francisco	Edward	Angel
2	Databases	2017	1	MCGRAW-HILL	585625700	Rochester	John	Smith
3	Computer Networks	2017	1	PEARSON	585626800	New York	Andrew	Garfield
4	Computer Graphics	2013	5	GRUPO PLANETA	585625300	Princeton	Edward	Petersman

branch_id	branch_ name	address	Title	copies	date_out	due_date
10	Henrietta	Rochester	Databases	2	12-JAN-2017	18-JAN-2017
11	Pittsford	Rochester	Computer Networks	2	02-FEB-2018	15-FEB-2018
12	Canindagua	Rochester	Computer Graphics	1	01-MAR-2018	05-MAR-2018
13	Fairport	Rochester	Databases	3	04-JAN-2017	06-FEB-2017

Applying 2 Normal Form:-

- All attributes (Non-key columns) are dependent on the key - By splitting the table into the book table and publisher table, and library_branch table and a table containing the library branch_id, title, copies and due-date & date-out we satisfy 2NF condition.

book_id	title	publishing_ year	years_since_ publishing	first_name	last_name
1	Databases	2017	1	John	Smith
2	Computer Networks	2017	1	Andrew	Garfield
3	Computer Graphics	2013	5	Edward	Petersman

publisher_ name	phone	address
MCGRAW-HILL	585625700	Rochester
PEARSON	585626800	Chicago
GRUPO PLANETA	585625300	Princeton

branch_id	branch_ name	address
10	Henrietta	Rochester
11	Pittsford	Rochester
12	Canindagua	Rochester
13	Fairport	Rochester

branch_id	Title	copies	date_out	due_date
10	Databases	2	12-JAN-2017	18-JAN-2017
11	Computer Networks	2	02-FEB-2018	15-FEB-2018
12	Computer Graphics	1	01-MAR-2018	05-MAR-2018
13	Databases	3	04-JAN-2017	06-FEB-2017

Applying 3 Normal Form:-

- All field (columns) can be determined by the key in the table and no other column - By splitting the Book table into the Publishing_Year table and Book_Author table we satisfy 3NF.

book_id	title
1	Databases
2	Computer Networks
3	Computer Graphics

book_id	first_name	Last_name
1	John	Smith
2	Andrew	Garfield
3	Edward	Petersman

publishing_ year	years_since_ publishing
2017	1
2016	2
2015	3

publisher_ name	phone	address
MCGRAW-HILL	585625700	Rochester
PEARSON	585626800	Henrietta
GRUPO PLANETA	585625300	Princeton

branch_id	branch_ name	address
10	Henrietta	Rochester
11	Pittsford	Rochester
12	Canindagua	Rochester
13	Fairport	Rochester

branch_id	Title	copies	date_out	due_date
10	Databases	2	12-JAN-2017	18-JAN-2017
11	Computer Networks	2	02-FEB-2018	15-FEB-2018
12	Computer Graphics	1	01-MAR-2018	05-MAR-2018
13	Databases	3	04-JAN-2017	06-FEB-2017

Applying 4 Normal Form:-

- No multivalued dependency - By splitting the library information table containing branch_id, title, copies, date_out and due_date into Book_Copies table and Book_Lending table we satisfy 4NF.

book_id	title
1	Databases
2	Computer Networks
3	Computer Graphics

book_id	first_name	Last_name
1	John	Smith
2	Andrew	Garfield
3	Edward	Petersman

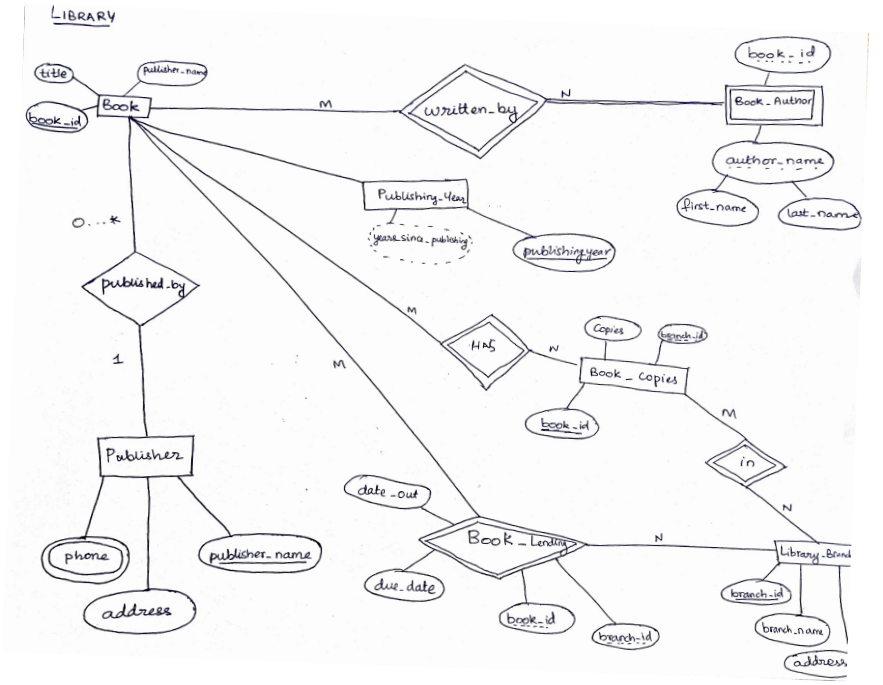
publishing_ year	years_since_ publishing
2017	1
2016	2
2015	3

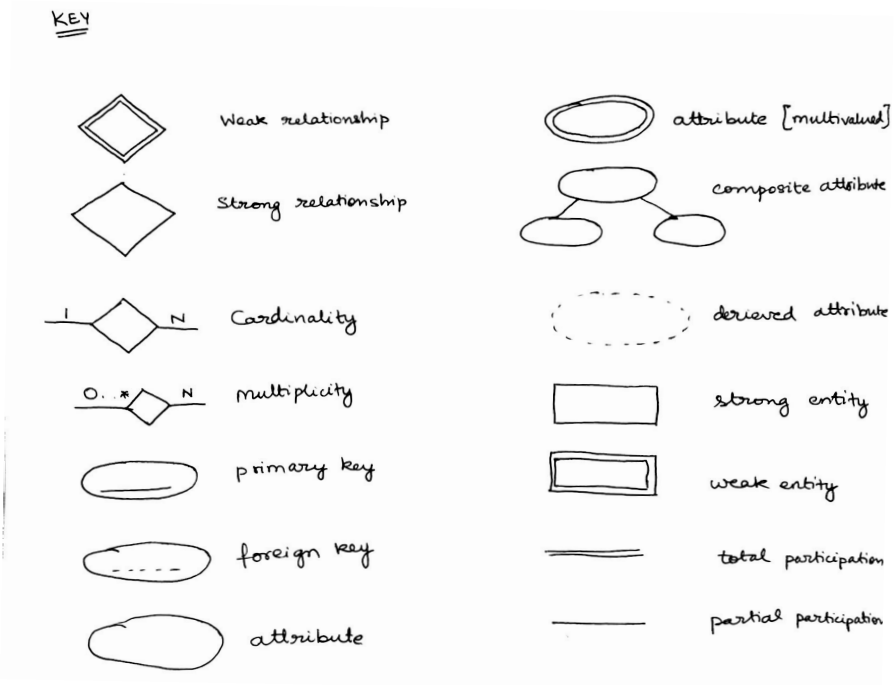
publisher_name	phone	address
MCGRAW-HILL	585625700	Rochester
PEARSON	585626800	Henrietta
GRUPO PLANETA	585625300	Princeton

branch_id	book_id	copies
10	1	2
11	2	2
12	3	1
13	1	3

branch_id	book_id	date_out	due_date
10	1	12-JAN-2017	18-JAN-2017
11	2	02-FEB-2018	15-FEB-2018
12	3	01-MAR-2018	05-MAR-2018
13	1	04-JAN-2017	06-FEB-2017

Entity-Relationship Diagram





The figure above has an entity relationship diagram for the database. Some of the key features we observe intend to observe include:-

- **Entity -** represented by means of rectangles, named with the entity set they represent
    - **Strong Entity Set -** An entity set that has a primary key is called as Strong entity set. Eg. Book
    - **Weak Entity Set -** The entity set which does not have sufficient attributes to form a primary key is called as Weak entity set. Eg. Book_Author

- **Attributes -** Attributes are the properties of entities. Attributes are represented by means of ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle)
    - **Composite attributes -** author_name in Book_Author table
    - **Single valued attribute -** book_id in Book table
    - **Multivalued attribute** - phone_number in Publisher table
    - **Derived attribute** - years_since publishing in Publishing_Year table can be derived from publishing_year

- **Cardinality -** Cardinality defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set
    - **One-to-many relationship -** Publisher and Book
    - **Many-to-Many relationship -** Book and Book_Author

- **Multiplicity -** defines the number of entities participating in the relationship
    - Book and Publisher defines a relationship where 1 publisher may publish 0 or more books

- **Participation constraints**
    - **Total Participation -** Each entity is involved in the relationship. Eg. Book_Author and written_by
    - **Partial Participation -** Not all entities are involved in the relationship. Eg. Publisher and published_by

- **Relationships -** Relationships are represented by diamond-shaped box. All the entities (rectangles) participating in a relationship, are connected to it by a line.
    - **Strong Relationship -** Each entity is involved in the relationship can exist independently. Eg. Publisher and Book can exist independently
    - **Weak Relationship -** Each entity is involved in the relationship cannot exist independently. Eg. Book and Book_Author

- **Relationship attributes -** relationship with attributes. For eg., Book_Lending table has attributes of date_out and due_date which define the date the book is lent and the due date.

Conclusion

In conclusion we see that we have a database that can be used to track books in a library. The key learning from this assignment include understanding the concept of E-R diagram and normalization. The E-R diagram helps us understand the relation between the various entities that are a part of our database. Normalizing the tables makes the database easy to understand and improves readability. It also implements the concept of separation of concerns by having each table handling its own specific function.

References

- Wikipedia
  https://en.wikipedia.org/wiki/Database_normalization
  https://en.wikipedia.org/wiki/Database

- Tutorial Point
  https://www.tutorialspoint.com/dbms/er_diagram_representation.htm