Database Design and Tech Survey

Kangdong Yuan

Table of content

**Pages**

1. **Project Introduction 3**
2. **Task 1: Requirement Analysis 3**
3. **Task 2: Conceptual Database Design 7**
4. **Task 3: Technology Survey 11**
5. **Task 4: Logical database design and Normalization 15**
6. **Citation list 17**

**Introduction**

The project entails the design, implementation, documentation and presentation of a database-backed web application for an online bookstore. Phase 1 of this project is presented in this document and focuses on the analysis and design of the envisioned web-application.

**Task 1: Requirement Analysis**

The proposed online bookstore system will aid in managing books, customers, books inventory and keep track of orders done by customers. Additionally, the system will have features to enable customers to give feedbacks on specific books; in form of a rating and a comment as well as rate feedback by other users. The rating on a user's feedback will be ranked as very useful, useful or useless. A user can also give an opinion on whether another user is trusted or not-trusted.

**Functional Requirements of the System**

* **Customer Registration**: The system will support registration of customers where the customer's details are captured, along with their unique username and a password.
  + **Data Required and integrity constraints:** A customer table will be created with the attributes;
  + **Customer**(customer\_id,name,email,password,address,phone,regdate)
    - **Customer\_id:** a unique system-generated identifier for each customer
    - **Email address:** this will be used as the login username; thus, every email has to be unique
    - **Password:** a password that meets minimum password requirements
* **Multiple Managers:** the system should have a means of allocating rights and permissions on the various modules; this will enable users to access various features of the system depending on their user level and user category.
  + **Data required:** a user’s category table will be created with category id; a unique identifier and category name. Each user will be put in a category.
    - **User\_Category** (category id, category name)
    - **User** (User ID, staff name, category id, address, phone, email)
  + A second table will be a permissions table; with permission ID, permission
    - **Permissions** (permission id, description)
  + A third table will link the users table with the permissions table, the permission table will have attributes; permission id, user id: The two IDs are foreign keys, the presence of a record with the user and permission will mean that the given user is allowed to that specific permission.
    - **User\_permission**(user\_id,permission\_id)
* **Ordering:** The system should allow registered users to place an order. After placing an order, the total amount should be shown to the customer. A customer will be able to order as many books as they wish and as many copies of a given book as they desire.
  + **Order** (order id, order\_date, customer\_id, order\_status)
  + **Order\_details** (detail\_id, order\_id, **ISBN**, quantity, purchase\_price, discount)
* **New Book Registration**: The system will enable store managers to add new books where they will record the details of the book and number of copies.
  + **Book (ISBN, book\_title, author, publisher, language, pub\_date, no\_pages, quantity, price, keywords, subject)**
* **Stock Addition:** this feature will enable store managers to add stock to the books already registered on the system.
  + To keep track of added stock, a table is required that records any additions or removal before altering the quantity in the book table
  + **Book\_movements(**ISBN,date, quantity)
* **Comments:** this system feature will enable registered customers to place a comment/edit a comment or delete a comment on a given book. The customer will be able to rate the book as well as write a short comment. The system will then store the comment with details such as rating, time and date and the comment.
  + **Comment** (customer\_id, **ISBN,** comment\_timestamp, rating, comment)
  + To ensure only one comment per book per customer, the customer id and ISBN will be used as the primary key; with a unique constraint on them, thus ensuring only one entry of the combination.
* **Usefulness rating**: the system should allow a user to rate the usefulness of other users’ comments; this rating should be one of the three; very useful, useful, or not useful.
  + **Usefulness\_rating**(**customer\_id, ISBN**, rater\_customer\_id, rating, date)
  + **customer\_id, ISBN:** this is a foreign key referencing the comment table, the ID of the person rating is stated as rater\_customer\_id
* **Trust Recording:** the system should allow users to declare other uses as either trusted or untrusted.
  + **TrustRecord**(record\_id, customer\_id, target\_customer\_id, status)
  + Both **customer\_id** and target\_customer\_id are foreign keys, referencing the customer table. The customer\_id is the ID of the person placing the rating while the target\_customer\_id is the person being rated.
* **Book browsing**: the system should enable users to search for books by various search metrics such as by; language, author, publisher and/or title.
  + This functional requirement uses existing books, author and publisher tables; as such no new database table is required to fulfil this requirement.
* **Useful Comments:** the system should allow a customer to get a specified number – between 5 and 10 - of top useful comments. The system should rank the usefulness of a comment by its average useful score.
* **Buying suggestions:** the system should display buying suggestions, based on a customer’s purchase. The suggestion should be derived from historical purchases data, by checking combination of book that previous customers have bought together. The system should order the book by purchase amount in a decreasing order.
* **Degrees of separation:** the system should be able to retrieve list of books written by authors who have 1 or 2 degrees of separation from the author or author of the book specified by the customer.
  + This functional requirement will be met by tables created, particularly the book, authors and publisher tables**.**
* **Book Statistics**: The system should enable the store manager get analytics report such as most popular books, popular authors and list of most popular publishers for a given quarter.
* **User awards**: The system should enable the manager to retrieve list best users based on trust and usefulness scores.

**Additional Functional and Non-functional Requirements**

* **Security:** the system should put in place measures to secure the data. This should be taken into consideration during coding such as cleaning data to avoid SQL injection and encrypting critical data such as passwords.
* **Ease of Use**: (non-functional requirement) the system should be designed in a way that makes it easy for users to easily access the various features, without needing extensive training on how to use the system.
* **Stock Management**: additional stock management features are required such as ability to remove damaged books from inventory.
* **Reporting Module**: additional reports such as inventory status report, inventory value and breakages reports.

**Task 2: Conceptual Database Design**

From the analysis of the case study, the database will have 12 tables, namely:

1. **Customer**(customer\_id,name,email,password,address,phone,regdate)
   * + **Customer\_id:** a unique system-generated identifier for each customer
     + **Name**: full name of the customer
     + **Email address:** this will be used as the login username; thus, every email has to be unique
     + **Password:** a password that meets minimum password requirements
     + **Address:** the customer’s address
     + **Phone:** the phone number
     + **Regdate:** the date the customer was resisted on the system
2. **User\_Category** (category id, category name)
   * Category id: a system generated identifier for the user category
   * Category name: name of the given category
3. **User** (UserID, staff name, category id, address, phone, email, password)
   * UserID: system generated primary key for the user
   * Staff name: full name of the staff member
   * Category id: the category the user belongs to
   * Address: the staff’s address
   * Phone: phone number of the staff member
   * Email: email address, which will also be used as the login username
   * Password: a login password
   * Permissions table will contain all permissions that can be issued to users
4. Permissions (permission id, description)
   1. Permission id: system generated primary key that uniquely identifies a permission
   2. Description : the permission description
5. User\_permission(user\_id,permission\_id)
   1. User\_id: foreign key; references permissions table
   2. Permission\_id : foreign key, references permission
6. Order (order id, order\_date, customer\_id, order\_status, user\_id)
   1. Order\_id: system generated unique primary key
   2. Order\_date:date of the order
   3. Customer\_id: a foreign key referencing the customer table
   4. Order\_status: the current status of the order
   5. User\_id: the id of the staff who processed the order
7. Order\_details (detail\_id, order\_id, ISBN, quantity, purchase\_price, discount)
   1. Detail\_id: primary key, auto generated
   2. Order\_id: foreign key referencing the order table
   3. ISBN: foreign key, referencing the book table
   4. Quantity: number of books purchase
   5. Purchase\_price: the price of the book
   6. Discount: amount of discount given
8. Book (ISBN, book\_title, author, publisher, language, pub\_date, no\_pages, quantity, price, keywords, subject)
   1. ISBN: primary key, unique for every book
   2. Book title: the title of the book
   3. Publisher: the book’s publisher
   4. Language: language of the book
   5. Pub\_date: date of publication
   6. No\_pages: number of pages
   7. Quantity: amount of stock for a given book
   8. Price: the set price of the book
   9. Keywords: keywords that can be used to search the book
   10. Subject: the book’s subject area
9. Book\_movements(log\_id,ISBN,logdate, quantity): this table keeps a transactions log of the books
   1. Log\_id: auto generated primary key
   2. ISBN: ISBN of a book
   3. LogDate: date a movement happened; either addition or substation
   4. Quantity: number of books in a transaction
10. Comment (customer\_id, ISBN, comment\_timestamp, rating, comment)
    1. Customer\_id: foreign key referencing the customer table
    2. ISBN: a foreign key referencing the book table
    3. Comment: the comment by the customer on the book
    4. comment\_timestamp: the date and time of the comment
    5. Rating: a rating of the book
11. Usefulness\_rating(customer\_id, ISBN, rater\_customer\_id, rating, date)
    1. Customer\_id: foreign key referencing the customer table;
    2. ISBN: a foreign key referencing the book table
    3. rater\_customer\_id: customer ID of the customer rating the comment
    4. rating: rating given for the comment
    5. date: date and time of the rating
12. TrustRecord(record\_id, customer\_id, target\_customer\_id, status)
    1. Both **customer**\_id and target\_customer\_id are foreign keys, referencing the customer table. The customer\_id is the ID of the person placing the rating while the target\_customer\_id is the person being rated.
    2. **Status**: the rating given

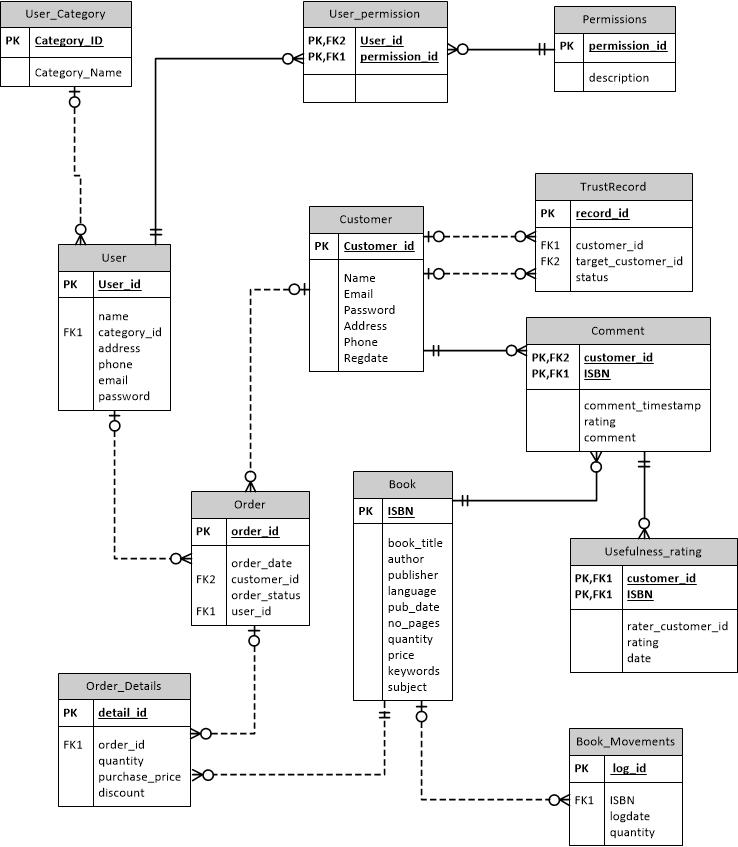


Figure 1.0 Conceptual Database Design

**Task 3: Technology Survey**

IT industry is developing at a very high rate. Knowledge of emerging technologies as well as top technologies in the market is critical in maintaining competitiveness. For web programming, numerous technologies, platforms and frameworks are in use. Key among these technologies include Python Django, Python Flask, PhP Laravel, and various Javascript frameworks. On the database side, leading database systems includes Oracle Databases, Microsoft SQL Server, SQLite and MySQL.

1. Frontend

The Frontend has many choices for us to use, there are three mainstreams in Frontend: HTML, CSS, JavaScript (Khrupa, 2020). And, each of them has its own advantage and disadvantage. If we want to find which one is best language for this project, we need compare the features of each languages.

The advantages of CSS (Tin.M, 2019):

* It is easy for beginner to use and learn.
* It has complete user interface.
* It is easy to do maintenance.
* It has good stability.

The disadvantages of CSS (U, 2020):

* It has too many restrictions on grammars.
* It lacks personalized outlook.
* It needs more time to write CSS code.

The advantages of JavaScript (college, 2020):

* JavaScript is very fast because any code functions can be run immediately instead of having to contact the server and wait for an answer.
* JavaScript is relatively simple to learn and implement.
* JavaScript plays nicely with other languages and can be used in a huge variety of applications.

The disadvantages of JavaScript:

* Because the code executes on the users' computer, in some cases JavaScript can be exploited for malicious purposes.
* JavaScript is sometimes interpreted differently by different browsers.

The advantages of Html (lab, 2020):

* It supports the latest versions of major browsers.
* It allows simple interactivities without external plugins.
* It is easy to learn and write.

The disadvantages of Html:

* It has many restrictions on multimedia.
* It has security concerns with web apps.

For me as a new learner of frontend language, I choose html to use, because it is easy to learn and it support my browser. Although it has some disadvantages on multimedia and web app, I don’t need to involve these respects in my project.

2. Backend

There are two backend languages that I have learned, which are python and PHP. So, I need compare the pros and cons of each languages to choose the better one.

The advantages of python:

* Python is easy to learn and write.
* Python don’t need compile.
* Python has many third-party packages to use.

The disadvantages of python:

* The running efficient is very low.
* The python environment is hard to build.

The advantages of PHP (college, 2021):

* PHP is a quick and easy server-side scripting language for web development and general use.
* PHP is a large community, widely used.
* PHP works well with databases, file systems, images.

The disadvantages of PHP:

* PHP is not optimized for desktop apps.
* PHP runs slightly slower than other programming languages.

Finally, I will choose python as backend language for this project, because I familiar with python and python has many packages to use.

3.Database

There are two choice for database in our project, MySQL or SQLite. So, I will compare the pros and cons of these database to choose the better one for our project.

The advantages of MySQL (S, 2019):

* MySQL provides a lot of database related features.
* MySQL provides good speed and performance.
* MySQL gives good user management and multiple access controls.

The disadvantages of MySQL:

* Requires some technical expertise to setup.
* Slightly different syntax as compared to conventional SQL.

The advantages of SQLite(S, 2019):

* File-based and easy to set up and use.
* Suitable for basic development and testing.
* Uses standard SQL syntax with minor alterations.

The disadvantages of SQLite:

* Lacks user management and security features.
* Not easily scalable.
* Is not suitable for big databases.

Finally, I choose SQLite as my database, because I am new to the develop environment. Setting up MySQL is very hard for me.

**Task 4: Logical database design and Normalization**

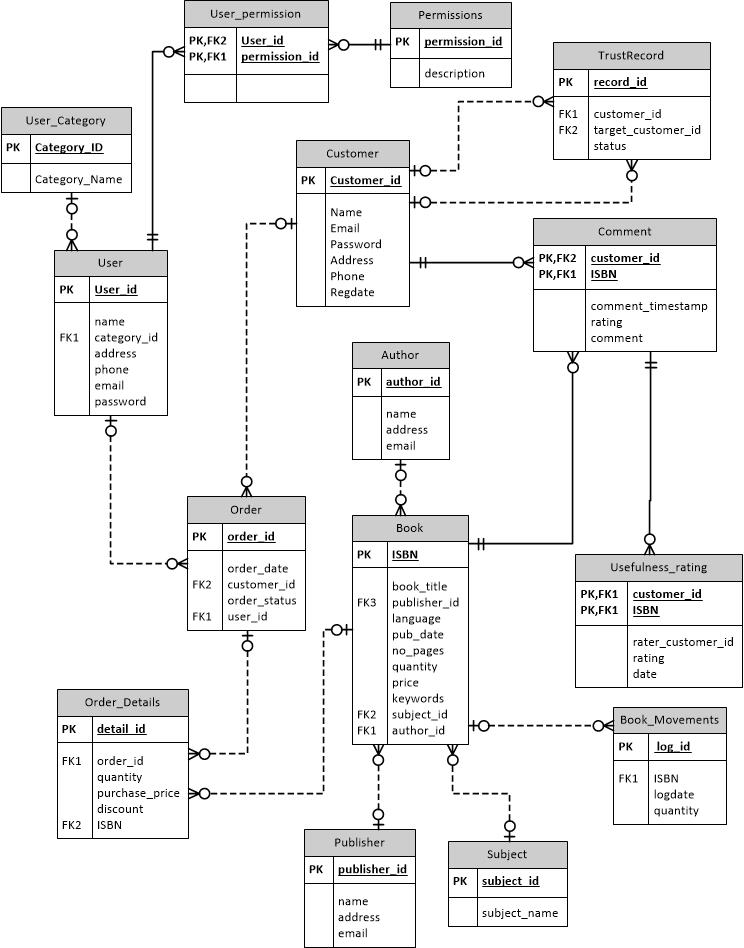


Figure 2.0 Logical database design and Normalization

Database Normalization

The Conceptual database design was normalized to BCNF (Boyce-Codd Normal Form). This was achieved by normalizing the Book table, where three extra tables; Publisher, Subject and Author tables were created. The creation of the extra tables eliminated duplicate data and ensured the database attains the desired normal form.

In order to attain the BCNF form, the following steps were observed;

1st Normal Form: All attributes are single valued and have unique names; all tables in the ER diagram had attained this state at the start of normalization, because no column had multiple data in one column.

2nd Normal Form: Tables have a single column primary key and all non-key attributes are fully dependent on the primary key. This was achieved by the creation of the 3 extra tables, thus eliminating redundancy

3rd Normal Form: Tables do not have transitive functional dependency. The books table had transitive dependencies, which were eliminated by creating tables like publisher and subject.

BCNF: Tables do not have more than one candidate key

**Citation list**

Khrupa, A. (2020, August 25). Top frontend development frameworks in 2018 for the amazing interfaces. Retrieved March 22, 2021, from <https://qarea.com/blog/best-frontend-programming-languages-to-create-beautiful-and-fast-interfaces>

Tin.M. (2019, November 15). Pros and cons of using a css framework. Retrieved March 22, 2021, from <https://medium.com/@ustinvaskin/pros-and-cons-of-using-a-css-framework-21ea7222b42e>

U. (2020, February 16). Pros and cons of using a css framework. Retrieved March 22, 2021, from <https://speckyboy.com/discussing-the-pros-and-cons-of-using-a-css-framework/>

College, M. (n.d.). Advantages &amp; disadvantages of javascript. Retrieved March 22, 2021, from <https://www.mediacollege.com/internet/javascript/pros-cons.html>

Lab, I. (2020, July 15). Using the pros and cons of html5 to enhance e-learning. Retrieved March 22, 2021, from <https://blog.commlabindia.com/elearning-development/pros-and-cons-of-html5>

College, I. (n.d.). The pros and cons of programming with PHP and .NET: Informatics Inc. Retrieved March 24, 2021, from <https://www.informaticsinc.com/blog/september-2018/pros-and-cons-php-and-net/amp>

S, E. (2019, March 27). What to use – sqlite or mysql? In-depth analysis for your convenience. Retrieved March 24, 2021, from <https://www.hostinger.com/tutorials/sqlite-vs-mysql-whats-the-difference/#:~:text=SQLite%20is%20a%20server%2Dless,to%20interact%20over%20a%20network>.