Library Database Management System

Nusan Adhikari

CSC675-03 Databases Systems

Id: 918392558

Github: adnusan

Milestone / Version	Date
M2 V1	April 19,2022
M1 V1	March 15, 2022

Table of Contents

LIBRARY MANAGEMENT SYSTEM

Section I: Project Description

Library Management System is software designed to do make the everyday functions of a library much easier and more efficient. This system will help the library to manage the inventory, keep track of in/out books, manage memberships/customers of the library, and detailed information about the book like an author, published date, etc. This system will be able to keep a record of which books are being borrowed, purchased, returned, lost by which member with their due dates. Additionally, this system will also have features like inventory management which will allow librarians to add a new book into the system and keep track of lost books if we need to add them.

This system also makes it easier for customers to borrow books as well as return them. This system will have a search function where people can search for the books and check if it's available to borrow or check the next available date. Additionally, this system will have a feature that will remind customers by email/phone about the due date to return books and allow customers to extend due dates for an extra fee from their online accounts. Features like sending reminder about due date could help customers from paying fine which could attract many customers to the library that uses this system. Furthermore, this system will have in store pickup feature where customer will be able to checkout online and pickup book in store which saves customer time and avoid contacts. In our current covid situation, many people are trying to avoid as much contact as possible so this could attract many customer who are conscious about the covid and health. This could also mean we can operate library even during pandemic when we cannot have customers inside library.

Section II: Use Cases

1. **Use Case:** Difficult managing borrowed/returned books in a library

   Actor: Diego(Librarian), Books, Customers

   Description: Diego is a librarian whose job is to rent out books to customers and take the returned books from customers. Diego keeps records of everything manually in an excel sheet, there is always a case where he loses track of the books he rented out/returned books, additionally, it is very timeconsuming to record everything manually and check if the customer returned book is past due date.

   This Library Management System will help Diego to rent out books and take returned books and keep track of which customers returned/borrowed books easily in software. Diego doesn't have to keep records manually, this system will scan the book and customer and mark it as borrowed or returned automatically with respective due dates and update the inventory. This system will let Diego know if the returned book was past due and he needs to charge extra fees to the customer.

2. **Use Case:** hard to find if books are available to borrow

   Actor: Jose(Customer), Books, Diego(Librarian)

   Description: Jose is in a library and he wants to borrow a certain book. The library is very big and has multiple floors of books. Jose is frustrated that he cannot find if the library has the book he is looking for. Jose asks the librarian whose name is Diego; Diego goes to look for the book on the floor, Diego finds out that the book Jose wants is being borrowed by someone else. Jose asks Diego when that book will be available to borrow and Diego takes time to find out about the next available date on his excel sheet.

Jose and Diego are both frustrated that it took so much time to find out that book was not available to borrow that day.

This Library Management System will have a search feature where customers like Jose or librarians like Diego will be able to search for a specific book and it shows the details about the books and if it is available to borrow or the next available date to borrow. It only takes a few clicks to find out about the books and saves time for both Jose and Diego.

3. **Use Case:** long process of checking out books

   Actor: Librarian, Jose(Customer), Books

   Description: Jose is a customer who found the book he wants to borrow. Jose is a regular customer of the library and he borrows books from the library very often. Jose is annoyed that he has to fill out the form and put his information every time he wants to borrow a book. Because customers have to enter personal information every time, it could leak to the public. Additionally, Since Jose borrows a lot of books, sometimes he mixes up or forgets the return due date and he has to pay a fine.

   This system will minimize the chance of getting personal information leaked as well as minimize the time to checkout/return book. This system will have a membership function that will allow librarians to create or manage membership for the customers. When customers sign up for the membership, they will only have to enter personal information once and they will be given a unique id. Customers like Jose can use that id to checkout books faster and he doesn't have to enter information every time he borrows a book. This system will also have a feature that will remind Jose about the return due date for books he borrowed by email or phone number.

4. **Use Case:** online order and pickup

   Actor: Dan (customer), Librarian, Books

   Description: Dan is a customer who knows which book he wants to borrow. Dan already checked it online if that book is available; he knows that book is available but he wishes he could just pay online to save time and avoid standing in a queue.

   This Library Database Management system will allow customers to reserve a book online and pick it up instore. Dan will be able to browse books, check out and pay online; when he walks into the store he can just pick up that book instead of waiting in line to pay. After Dan checks out the book online, Librarian can see the online order and they will get the book ready to be picked up, this saves time for both customer and the Librarian. Within a few days of implementing this feature, Librarian could see many people use the pickup feature which saves him time because he doesn't have to interact with each customer for too long.

5. **Use Case:** online payment system

   Actor: Dan(Customer), Librarian, Books

   Description: When customers like Dan checkout online, he wishes they had other payment options like crypto or PayPal. The option to pay from PayPal will be faster because PayPal had a oneclick checkout feature. Dan is frustrated that he has to enter his debit/credit card every time he has to pay.

   This system could implement Paypal and crypto features where customers can use those to check out faster. This could save time for customers since they don't have to enter every information every time they checkout. This feature will allow encourage users to check out more books because it's much easier.

Section III: Database Requirements

1. General User
    a. General User is an unregistered, registered user or librarian
    b. General User is a premium user
    c. General users shall register only one account with a unique email and phone number
    d. General User shall be able to search many books

2. Roles
    a. Role shall be used by many users

3. Librarian
    a. Librarian is a general user
    b. There shall be many Librarian
    c. Librarian is premium members
    d. Librarian shall be one and only to edit roles
    e. Librarian shall be one and only edit/add/remove books from inventory
    f. Librarian shall be one and only to block members' account
    g. Librarian shall be one and only to unblock members' account
    h. Librarian shall be one and only to fine members
    i. Librarian shall be able to take many returned books

4. Registered Members
    a. Registered Members are a general user
    b. Registered Members shall have at least one default payment
    c. Registered Members shall have at least one address, phone, email, ID number
    d. Registered Members shall log into their account from many devices
    e. Registered Members shall be able to preview many books
    f. Registered Members shall rent many book at a time
    g. Registered Members shall be able to return book
    h. Registered Members shall be able to signup for only one premium membership

5. Premium Members
    a. Premium Members are a registered user
    b. Premium Members are general users
    c. Premium Members shall have many users
    d. Premium Members shall be able to search books
    e. Premium Members shall be able to preview many books
    f. Premium Members shall be able to check out many books
    g. Premium Members shall be able to return books
    h. Premium Members shall have at least one default payment
    i. Premium Members shall be able to cancel memberships
    j. Only Premium Members shall be able to order to pick up many books
    k. Only Premium Members shall receive notification about due date

6. Books
    a. There shall be many books
    b. Books shall have one or many authors

c. Books shall have many publishers
d. Books shall have at least one title, authors
e. Books shall have at least one published date
f. Each book shall have only one unique serial number
g. Book shall have zero or many categories
h. Book shall be previewed by many users, members
i. Book shall have only one stored location address
j. Book shall have many languages, publisher
k. There shall be many copies of the same book
l. Book can be rented by multiple members, users
m. Book shall have status if it's available or rented

7. Payment Method
    a. A payment method is a credit/debit card, PayPal, or cryptocurrency
    b. A cryptocurrency payment method only accepts BTC, ETH, or Dodge
    c. A cryptocurrency payment method is linked to only one wallet address
    d. payment method can be linked to many premium accounts

8. Notification
    a. Notification shall have only premium members
    b. Notification shall have atleast one user email
    c. Notification shall have atleast one user phone number
9. Racks
    a. There shall be many racks
    b. Racks shall have many books
    c. Each rack shall have a unique number

d. Rack numbers is a addresses for books

10. Address

a. Members shall have multiple address
b. Address shall have one street number
c. Address shall have one zipcode
d. Address shall have one state
e. Address shall have one country
f. Address shall have one city
g. Many Address shall have one member

11. Computers

a. Library shall have many computers
b. General users shall use many computers
c. Registered members shall use many computers
d. Premium members shall use many computers
e. Each computers shall be used by one users at a time

12. Fax

f. Library shall have many fax machine
g. General users shall use many fax machine
h. Registered members shall use many fax machine
i. Premium members shall use many fax machine

13. Printers

j. Library shall have many printers

k. General users shall use many printer

l. Registered members shall use many printer

m. Premium members shall use many printer

14. Pick Up

a. Only premium members shall be able to pickup many books

b. Many premium members shall be able to pickup many books

15. Fine

a. Many users shall have many fine

b. Only librarian shall be able to charge many fine

c. Fine shall be calculated based on late days

16. Receipt

a. Receipt shall be received by one customer

b. Receipt shall be emailed

c. Receipt shall be printed

17. Rental

a. There shall be many rental

b. Rental shall have many books

c. Rental shall have many members

d. Rental shall have rented date

e. Rental shall have returned date

Section IV: Detailed List of Main Entities, Attributes, and Keys

1. User (Strong)
    - user_id: key, numeric
    - full_name: composite, alphanumeric
    - Address: weak key, multivalue, alphanumeric
    - email: alphanumeric
    - phoneNumber: weak key,multivalue, alphanumeric
    - Role: weak key, numeric
    - Government_id: weak key, alphanumeric

2. Book (Strong)
    - book_id: weak key, numeric
    - Serial_num: key, alphanumeric
    - title: simple, alphanumeric
    - Authors_id: key, alphanumeric
    - publication_date: multivalue, timestamp
    - Publisher_info: key, alphanumeric
    - Quantity:numeric

3. Role (Strong)
    - Role_id: key, numeric
    - Description: alphanumeric

4. Rented (weak)
    - Rent_id: key, numeric
    - Book_id: weak key, multiplevalue, numeric
    - User_id: weak key, numeric
    - Publisher_id: weak key, numeric
    - Rented_date: timestamp
    - Returned_date: timestamp
    - fine: weak key, multivalue, numeric
    - Due_date: derived, timestamp

5. Librarian (weak)
    - librarian_id: key, numeric
    - User_id: weak key, numeric
    - Role: weak key, numeric
    - Action_id: weak key, numeric

6. Registered Members (weak)
    - Member_id: key, numeric
    - User_id: weak key, numeric
    - Role:weak key, numeric
    - Rent_id: weak key, numeric
    - Payment_id: weak key, numeric

7. Premium Members (weak)
    - Member_id: key, numeric
    - User_id: weak key, numeric

- Role:weak key, numeric
- Pickup_id: weak key, multivalue, numeric
- Payment_id: weak key, numeric

8. Pick Up (weak)
    - Pickup_id: key, numeric
    - User_id: weak key, numeric
    - Book_id: weak key, numeric
    - Rent_id: weak key, numeric
    - Pickup_date: timestamp

9. Authors (strong)
    - Author_id: key, numeric
    - Author_name: alphanumeric
    - Author_DOB: timestamp
    - Author_deathDate: timestamp
    - Author_Gender: alphanumeric
    - Author_nationality: multivalue, alphanumeric
    - Author_languages: multivalue, alphanumeric
    - Author_Bio: alphanumeric

10. Membership (weak)
    - Member_id: key, numeric
    - User_id: weak key, numeric
    - Payment_id: weak key, alphanumeric

11. Publisher (strong)
    - Publisher_Id: key, numeric
    - Publisher_name: composite, alphanumeric
    - published_country: multivalue, aplhanumeric

12. Racks (strong)
    - Rack_id: key, alphanumeric
    - Rack_floor: numeric
    - Rack_column: alphanumeric

13. Payment (strong)
    - Payment_Id: key, numeric
    - Payment_type:weak key, numeric
    - Payment_amount: composite, numeric
    - User_id: weak key, numeric

14. Debit/Credit (weak)
    - debitCredit_id: key, numeric
    - Payment_type: alphanumeric
    - Payment_Id: weak key, numeric

15. Paypal (weak)
    - PayPal_id: key, numeric
    - Payment_type: alphanumeric
    - Payment_Id:weak key, numeric

16. Crypto (weak)

- Wallet_address:key, alphanumeric
- crypto_type: alphanumeric
- Payment_Id:weak key, numeric

17. Address (strong)

- Address_id: key, numeric
- Street: alphanumeric
- City: alphanumeric
- ZipCode: numeric
- State: alphanumeric
- Country: alphanumeric

18. Notification (weak)

- Notification_id: key, numeric
- User_id: weak key, numeric
- User_role: weak key, numeric
- User_email: derived, alphanumeric
- User_phone: derived, alphanumeric
- Rental_id: weak key

19. Printers (strong)

- Printer_id: key, numeric
- Print_pages: numeric
- Price: numeric

20. Fax (strong)

- Fax_id: key, numeric
- price:numeric

21. Computers (strong)

- Computer_id: key, numeric
- Uses_start: timestamp
- Uses_end: timestamp
- Price: numeric

22. Category (strong)

- Category_id: key, numeric
- Category_description: multivalue, alphanumeric

23. Fine (strong)

- Fine_id: key, numeric
- Fine_amount: derived, numeric

24. Government Id (strong)

- Gov_id: key numeric
- Id_number: alphanumeric

25. Phone Number (strong)

- Phone_id: key, numeric
- Area_code: numeric
- Numbers: numeric

26. Receipt (strong)

- Recipt_id: key, numeric
- user_id: weak key, numeric
- Date: timestamp
- Amount: numeric
- Book_id: weak key, numeric

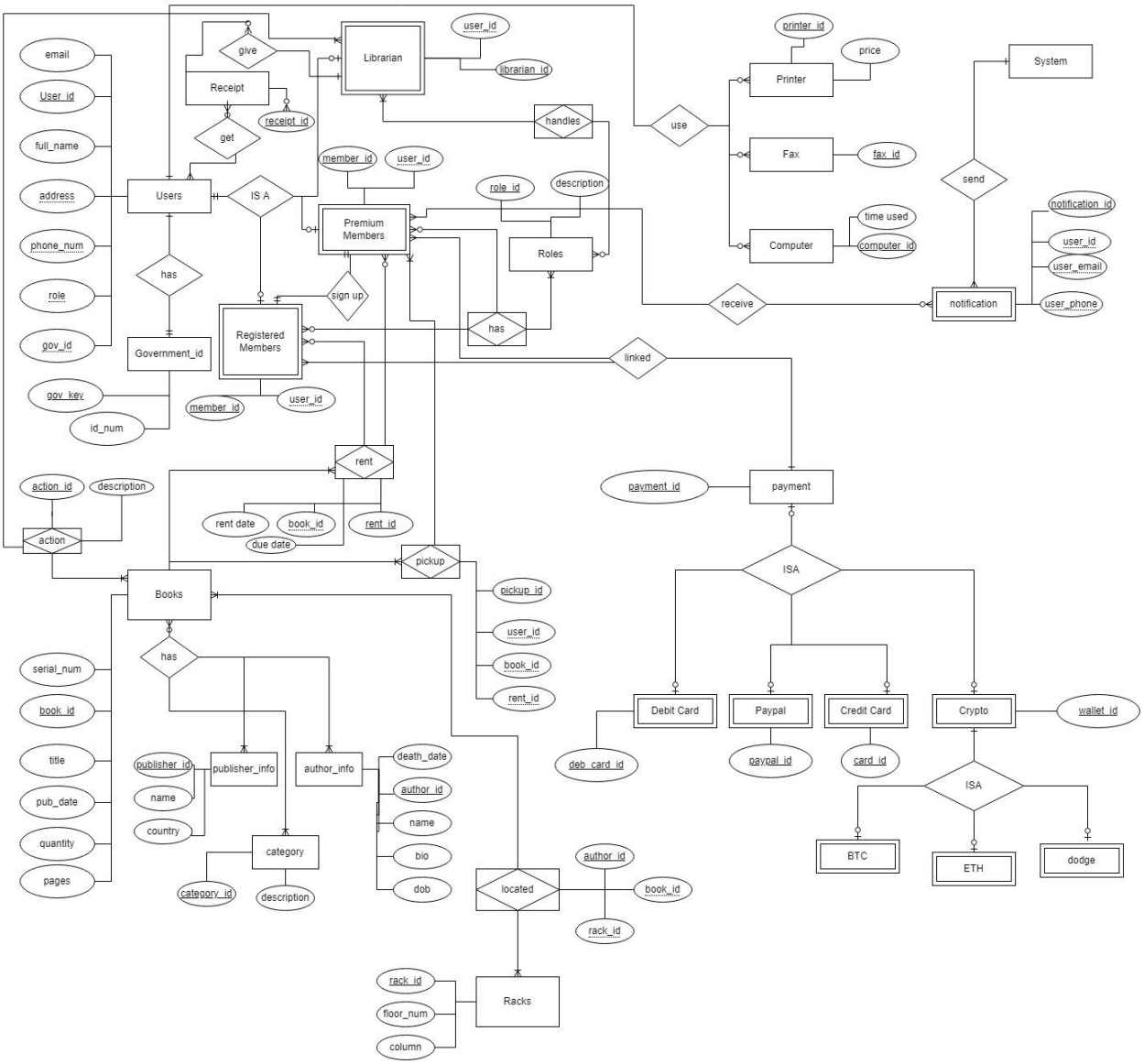
27. Membership Status (strong)

- Status_id: key, numeric
- Status_description: alphanumeric
- Exp_date: timestamp
- Start_date: timestamp

28. Actions (strong)

- Action_id: key, numeric
- Action: alphanumeric

Section V: Entity Relationship Diagram (ERD)



Section VI: Testing Table

Rule	Entity A	Relation	Entity B	Cardinality	Pass/Fail	Description
1	User	creates	Registered Account	1 to 1	pass	none
2	Registered Member	Sign up	Premium account	1 to 1	pass	
3	User	IS A	Registered Acc	1 to 1	pass	
4	User	IS A	Premium Member	1 to 1	pass	
5	User	IS A	Librarian	1 to 1	pass	
6	Librarian	Sign Up	Premium Acc	M to N	fail	Cannot sign up yet
7	User	has	Government ID	1 to 1	pass	
8	User	has	address	1 to M	fail	User only has one address
9	User	use	Printer	1 to M	pass	
10	User	use	Computer	1 to M	pass	
11	User	use	Fax	1 to M	pass	
12	Registered Member	rent	Books	M to N	pass	
13	Premium Member	rent	Books	M to N	pass	
14	Registered Member	pickup	Books	M to N	fail	Only premium member can use pickup feature
15	Premium Member	pickup	Books	M to N	pass	
16	Book	has	author	M to M	pass	
17	Book	has	Publisher info	M to M	pass	
18	Book	has	address	M to M	pass	

19	Registered Member	linked	payment	M to 1	pass	
20	Premium Member	linked	payment	M to 1	pass	

Section VII: Database Model/EER

I have included this image as eer.png inside milestone2 folder if you need to view this more clearly. Thank you :)

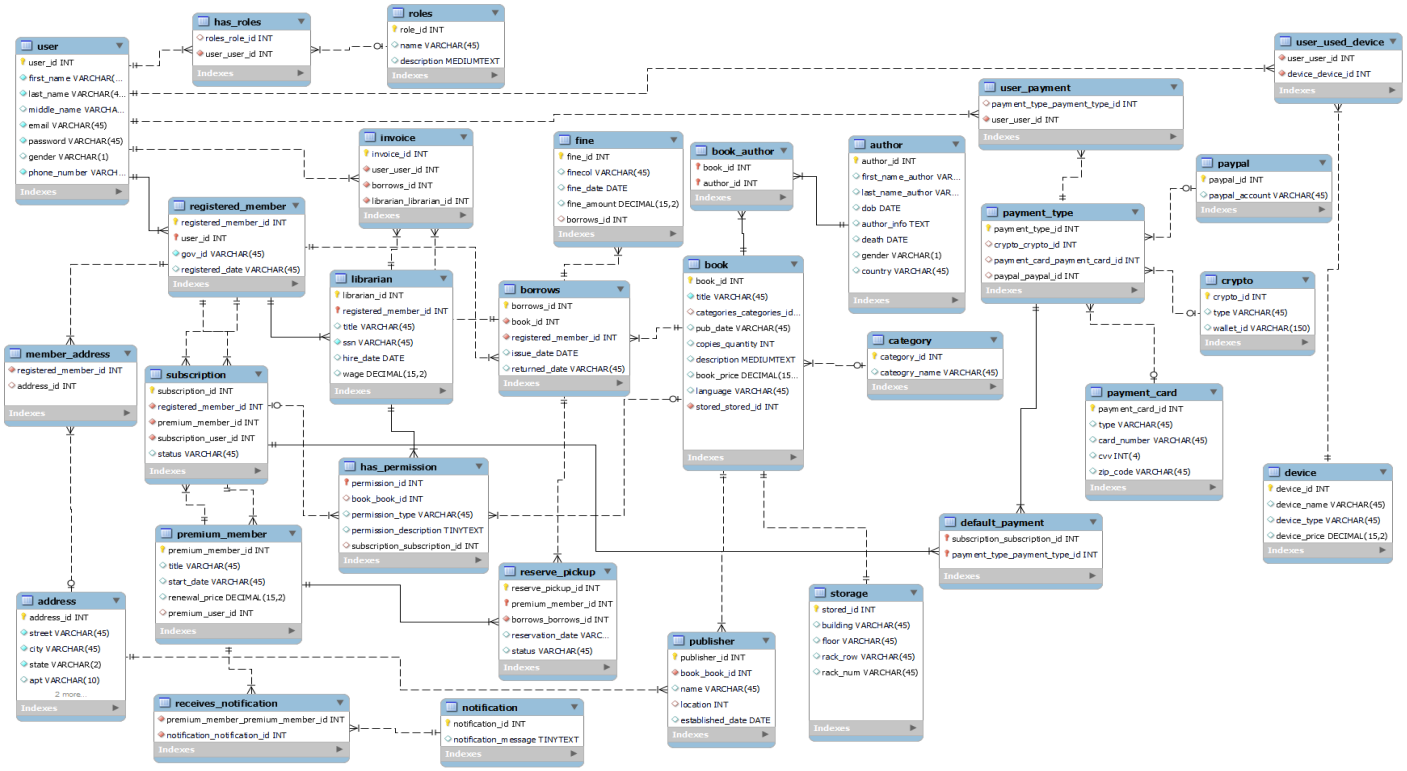


Table	FK	ON DELETE	ON UPDATE	Description
has_roles	roles_roles_id	SET NULL	CASCADE	If a role is deleted, a user holding that role will have no roles until assigned
has_roles	user_user_id	NO ACTION	NO ACTION	If a user is deleted, then nothing will happen to roles
registered_member	user_id	NO ACTION	RESTRICT	If a user is deleted, that user record will still be in our database. We will restrict user_id from changing because each user has a unique id that cannot be changed and auto-generated.
member_address	registered_member_id	NO ACTION	NO ACTION	If a registered member is deleted, nothing will delete from this table because we might still need the registered member address in the

				future.
member_address	address_id	SET NULL	CASCADE	If an address is deleted, then the user will have no address until entered again.
premium_member	premium_user_id	CASCADE	CASCADE	If a user_id is deleted then that user will no longer be premium member.
librarian	registered_member_id	CASCADE	CASCADE	If registered member is deleted, then the librarian with that member must be deleted
has_permission	book_book_id			If book is deleted, it doesn't affect permission table, other table still has permission
has_permission	librarian_id	CASCADE	CASCADE	If librarian is deleted, then permission of that librarian must be deleted
has_permission	subscription_id			Subscription doesn't affect this table
borrows	book_id	NO ACTION	NO ACTION	Even if book_id is deleted, user will still borrow the books
reserve_pickup	borrows_id	CASCADE	CASCADE	If borrow_id is deleted or updated, then reserve_pickup must be updated or deleted.
publisher	fk_publisher_address	SET NULL	CASCADE	If address is deleted, then publisher address will be set to null until new address is entered.
publisher	book_id	CASCADE	CASCADE	If a book is deleted, publisher_id related to that book will be deleted.
receives_notification	premium_member_id	CASCADE	CASCADE	If premium member is deleted, then that member should not receive notification.
receives_notification	notification_id	NO ACTION	NO ACTION	Nothing will happen if notification is deleted
book	category_id	NO ACTION	SET NULL	If category is deleted, then that book will have no category until entered.
book	stored_id		CASCADE	If stored place of book is updated, then it should be updated in book table as well
book_author	book_id	CASCADE	CASCADE	If book is deleted then table of

				book_author must be deleted because we no longer have that book
book_author	author_id	SET NULL		If author is deleted, then the book will have no author until reentered
user_payment	user_id	CASCADE		If user is deleted then payments of that user must be deleted
user_payment	payment_id	SET NULL		If payment is deleted, the user payment will have no payment type.
payment_type	crypto_id	SET NULL	CASCADE	If crypto table is deleted then crypto from payment_type will have no crypto until reentered.
payment_type	payment_card	SET NULL	CASCADE	If cardtable is deleted then crypto from payment_type will have no card until reentered.
payment_type	paypal_id	SET NULL	CASCADE	If paypal table is deleted then crypto from payment_type will have no paypal_id until reentered.
user_used_device	user_id	CASCADE		If user is deleted, table of devices used by that user must be deleted.

Section XI: Testing Table

Entiy	SQLQuery	Pass/Fail	Error Des	Possible Solution
user	delete	fail	Foreign key constrain fail	
user	update	fail	Cannot delete parent row	I have set user to be restrict in one table, need to fix that table
book	delete	fail	Cannot delete parent row	Same problemas above table
book	update	pass		
address	delete	pass		
address	update	pass		
author	delete	pass		
author	update	pass		
book_author	delete	pass		
book_author	update	pass		
borrows	delete	fail	borrow _id cannot be deleted to keep record for invoice and order status.	
borrows	update	pass		
category	delete	fail	Cannot delete parent row	Need to fix foreign key in table
category	update	fail	Wrong book id entered	Fixable by entering correct book_id from table
crypto	delete	pass		
crypto	update	pass		
default_payment	delete	pass		
default_payment	update	pass		
device	delete	fail	Error Code: 1451. Cannot delete or update a parent row	Might fix if set delete to cascade
device	update	fail	Foreign key constrain fail	Set update to cascade

fine	delete	pass		
fine	update	pass		
has_permission	delete	pass		
has_permission	update	fail	Cannot update child row	Need to set ON UPDATE to cascade
has_roles	delete	pass		
has_roles	update	fail	Misspelled roles_role_id	Fix attribute name
invoice	delete	pass		
invoice	update	pass		
librarian	delete	fail	Cannot delete parent	Set IN DELETE to cascade
librarian	update	pass		
member_address	delete	pass		
member_address	update	fail	Cannot add or update child	Set ON UPDATE to cascade
notification	delete	fail	Error Code: 1146. Table 'librarymanagementdb.notificatiion' doesn't exist	Fix notification spelling
notification	update	fail	Cannot delete parent row	Change Update restrict to cascade
payment_card	delete	pass		
payment_card	update	pass		
payment_type	delete	fail	Cannot delete parent	Set ON DELTE to cascade
payment_type	update	fail	Cannot update parent	Set ON UPDATE to cascade
paypal	delete	fail	Cannot delete parent	Set ON DELTE to cascade
paypal	update	pass		
storage	delete	fail	Cannot delete parent	
storage	update	pass		

registered_member	delete	fail	Registered member set restrict on delete	Need to set DELETE to Cascade
registered_member	update	fail	Restrict update to keep prevent users from changing id, but my logic is wrong	Need to cascade on update
user_used_device	delete	pass		
user_used_device	update	fail	Foreign key constrain fail	Need to change ON UPDATE to Cascade