

Project Report

Course: CS-GY 6083 Principles of Database Systems

Section Number: B

Date of Submission: May 7th

Name: Yilan Dong (yd2067), Zihao Chu (zc2069)

EXECUTE SUMMARY

- Business case

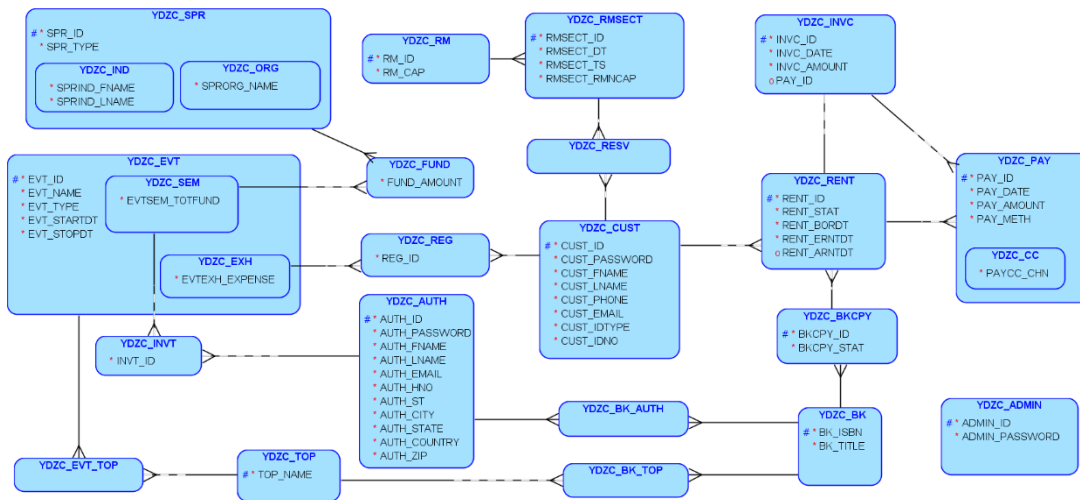
We support customer and author to sign up and provide services for both of them, administrators are responsible for management.

First, these three types of users are using the same page to login. So, we add a column to record users' type which is set in database using `<select><option></option></select>` when they sign up. And at the PHP file, based on this column, it uses different \$SQL to search in database. Also, at the sign-up page, we use JavaScript to set part of the input disabled so that customers or authors only submit relevant information.

Second, for customers they can borrow books, reserve a study room and register for an event (exhibition). And all of these three services, they can search through different columns. For example, search for a book can through ISBN, book title, or author name. Since author's name are stored in two columns, so we use concat() function. Also, in their home page, they can see all history information like all rent entries. For author, they can only see their book, and check for the seminar they are invited. Administrator has all access to all the tables, so they can search(select), edit(update), create and delete.

Third, to improve the speed, we implement index over some attributes such as sponsor type, event type, etc. Also, views are used for customers' services to ensure the data's security and avoid complex SQL statements. Meanwhile, we use procedures to simplify updating and inserting process, increase model security.

- Logical and Relational Model Design



We create a new table named "YDZC_ADMIN" to store administrators' information. Also, password column is added to "YDZC_CUST" and "YDZC_AUTH". Besides, we add a relation between "YDZC_INV" and "YDZC_PAY" so that we don't need to join multiple tables to get each payment's invoice.

CONFIGURATION

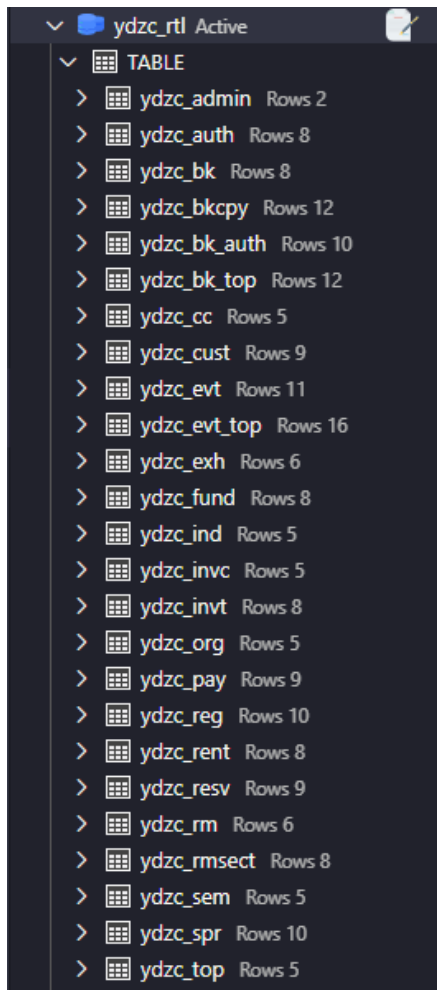
Browser: Edge, Safari, Chrome

Programming Language: HTML, CSS, JAVASCRIPT, PHP

Database: MariaDB 10.4.18

DATA

Our database is called ydzc_rtl, with 25 tables. We tried to set at least 5 instances for each table at initial and there would be more records at the demo since validating database could create a lot of records.



WEB APPLICATION SCREENSHOTS

Login Page

A screenshot of a login form. The form has a light blue header with the word 'Login'. Below the header, there are two input fields: 'Username' and 'Password'. Below the 'Password' field is a blue 'Sign in' button. At the bottom of the form, there is a link that says 'No account? create an account'.

Customer Home Page

read THINK learn

Home Book Event Study room Logout

ID	8
Name	Sasha Blouse
Email	sasha@nyu.edu
Phone	2129981218
Identification Type	Passport
Identification Number	437625008

Edit

Profile Edit Page

read THINK learn

Home Book Event Study room Logout

ID	8
First Name	<input type="text" value="Sasha"/>
Last Name	<input type="text" value="Blouse"/>
Email	<input type="text" value="sasha@nyu.edu"/>
Phone	<input type="text" value="2129981218"/>
Identification Type	<input type="text" value="P"/>
Identification Number	<input type="text" value="437625008"/>

Save Back

Book Search Page


read THINK learn

Home Book Event Study room Logout

ISBN Search

ISBN	Title	Author	Topic	Copy ID	Status
9782222211111	Western Culture	Zeke Yeager	Arts	1	Unavailable
				2	<input type="button" value="Available"/>
				3	<input type="button" value="Available"/>
9784444433333	Van Gogh: The Life	Annie Leonhart	Arts	5	Unavailable
9789999988888	Ancient Egypt	Falco Grice	Arts	12	<input type="button" value="Available"/>

Borrow Book




[Home](#) [Book](#) [Event](#) [Study room](#) [Logout](#)

ISBN	9788888877777
Title	Hundred Thousand Whys
Copy ID	10
Borrow Date	2021-05-07
Expected Return Date	2021-06-06


[Confirm](#) [Back](#)

Manage Page (Administator)



[Home](#) [Customer](#) [Author](#) [Book](#) [Event](#) [Study Room](#) [Topic](#) [Logout](#)

[Search](#)



Manage Customer


For CUSD

[Search Information](#)

[Edit Information](#)

[Delete Information](#)

Manage Customer Information (Administator)



[Home](#) [Customer](#) [Author](#) [Book](#) [Event](#) [Study Room](#) [Topic](#) [Logout](#)

[Search](#)

ID	first name	last name	phone	email	idtype	idno	edit	delete
1	Eren	Yeager	2129981211	eren@nyu.edu	S	437625001	edit	delete
2	Mikasa	Ackerman	2129981212	mikasa@nyu.edu	D	437625002	edit	delete
3	Armin	Arlet	2129981213	armin@nyu.edu	S	437625003	edit	delete
4	Reiner	Braun	2129981214	reiner@nyu.edu	P	437625004	edit	delete
5	Jean	Kirstein	2129981215	jean@nyu.edu	D	437625005	edit	delete
6	Marco	Bott	2129981216	marco@nyu.edu	S	437625006	edit	delete
8	Sasha	Blouse	2129981218	sasha@nyu.edu	P	437625008	edit	delete
9	Historia	Reiss	2129981219	historia@nyu.edu	D	437625009	edit	delete

SECURITY

- Password Encoding

We use Sha1 method to encode user's password and implement it in the Client side. After users submitting their sign-up form, we hash the password and get encoded version. And then send post message with all information to the Server side.

More specifically, we use jquery ('inputid').change() to realize it. When the input box of password changes, it will be captured and go to the function in .change() to hash the password.

We find that Sha1 is not safe enough, SSL performs better. However it's hard for us, so Sha1 is chosen.

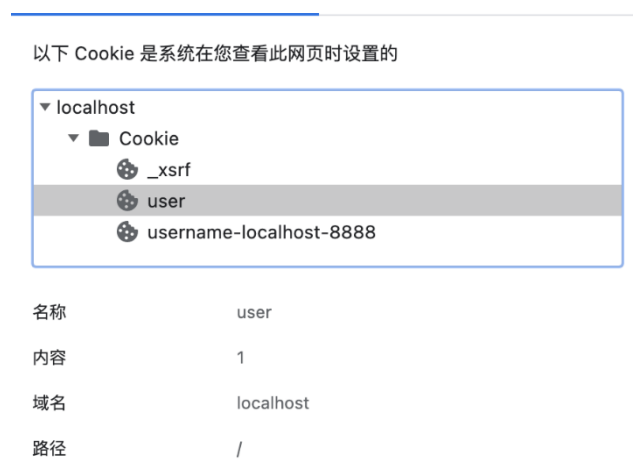
- SQL Injection

SQL injection may occur when someone type in <input></input> with special word like ", password='0' where cust_id='1'#" which can result in users' data modified.

We use htmlspecialchars function provided by PHP. Before we insert the input into SQL statement, we call htmlspecialchars(\$id, ENT_QUOTES) to flit apostrophe etc.

- Users' Session

We use cookie to keep user's session. After users sign in to the website, we can automatically generate a cookie. Cookie's name is "user" and value is users' ID, and since we only implement it locally, so cookie's path is "/".



Before the home page loading, we will check the cookie. So users can visit their pages without signing in again. After cookie expires, it will jump to sign in page. Also we can trace the current user's ID for some SELECT or INSERT operation happend in book study room server and so on.

REFLECTION

Through this project, both our frontend and backend skills are well practiced. Since this is the first time to develop frontend for both of us, one of the challenging jobs is to learn those well-developed and mature frameworks, as well as programming in js, php. Thus, time is indeed precious, and becomes our constraints.

This is our first-time cooperation, to improve the efficiency we use GitHub to manage our code but still we encountered some problems for not communicating in time. So, if we can have more time to discuss off-line, some problems won't occur.

BUSINESS ANALYSIS

(Q1) Table joins with at least 3 tables in join

```
Run SQL
1 SELECT concat(a.auth_fname, ' ', a.auth_lname) Author_Name, b.
   bk_title Book_Title, b.bk_isbn ISBN FROM ydzc_auth a, ydzc_bk
   b, ydzc_bk_auth c WHERE a.auth_id=c.auth_id AND b.bk_isbn=c.
   bk_isbn ORDER BY Author_Name, Book_Title;
2
```

		* Author_Name	* Book_Title	* ISBN
		Filter	Filter	Filter
	1	Annie Leonhart	Van Gogh: The Life	9784444433333
	2	Bertolt Hoover	Le Petit Prince	9783333322222
	3	Falco Grice	Ancient Egypt	9789999988888
	4	Falco Grice	Hundred Thousand Whys	9788888877777
	5	Gabi Braun	Hundred Thousand Whys	9788888877777
	6	Marcel Galliard	Lonely Planet	9785555544444
	7	Pieck Finger	Little Duck	9787777766666
	8	Porco Galliard	The Beauty of Science	9786666655555
	9	Zeke Yeager	Le Petit Prince	9783333322222
	10	Zeke Yeager	Western Culture	9782222211111

Explanation: We want to retrieve information of multiple-to-multiple relationship between authors and books by 3 tables (ydzc_auth, ydzc_bk, ydzc_bk_auth). Here, we have instances of each book with its each author separately.

(Q2)

```
Run SQL
1 SELECT a.top_name Topic FROM ydzc_top a, ydzc_bk_top b WHERE a.
  top_name=b.top_name AND b.bk_isbn IN (SELECT b.bk_isbn FROM
  ydzc_bk_auth a, ydzc_bk b WHERE 1=a.auth_id AND a.bk_isbn=b.
  bk_isbn);
2
```

<input checked="" type="checkbox"/>	<input type="checkbox"/>	* Topic	⬆
		Filter	
	1	Arts	
	2	Children	

Explanation: We want to show the topics set of all the books written by Author with auth_id=1.

(Q3)

```
Run SQL
1 SELECT evt_id Event_ID, evtsem_totfund Fund FROM ydzc_sem WHERE
  evtsem_totfund > (SELECT AVG(evtsem_totfund) FROM ydzc sem a,
  ydzc_evt b, ydzc_evt_top c WHERE a.evt_id=a.evt_id AND b.
  evt_id=c.evt_id AND c.top_name="Science");
2
```

<input checked="" type="checkbox"/>	<input type="checkbox"/>	* Event_ID	⬆	* Fund	⬆
		Filter		Filter	
	1	21006		1500.00	
	2	21007		2000.00	

Explanation: We want to show those seminar with fund higher than the average fund of seminar of Science topic.

(Q4)

Run SQL





```
1 SELECT a.auth_id Author_ID, f.top_name Topic FROM ydzc_auth a,  
ydzc_invt b, ydzc_sem c, ydzc_evt d, ydzc_evt_top e, ydzc_top f  
WHERE a.auth_id=b.auth_id AND b.evt_id=c.evt_id AND c.evt_id=d.  
evt_id AND d.evt_id=e.evt_id AND e.top_name=f.top_name  
2 UNION  
3 SELECT a.auth_id Author_ID, e.top_name Topic FROM ydzc_auth a,  
ydzc_bk_auth b, ydzc_bk c, ydzc_bk_top d, ydzc_top e WHERE a.  
auth_id=b.auth_id AND b.bk_isbn=c.bk_isbn AND c.bk_isbn=d.  
bk_isbn AND d.top_name=e.top_name  
4 ORDER BY Author_ID, Topic;  
5
```

<input checked="" type="checkbox"/>	<input type="text" value="Q"/>	* Author_ID ▾	* Topic ▾
		Filter	Filter
	1	1	Arts
	2	1	Children
	3	1	Science
	4	2	Children
	5	2	Science
	6	3	Arts
	7	3	Science
	8	4	Science
	9	4	Travel
	10	5	Children
	11	5	History
	12	5	Science
	13	6	Children
	14	6	Science
	15	7	Children
	16	7	History
	17	7	Science
	18	8	Arts
	19	8	Children
	20	8	History
	21	8	Science

Explanation: We want to find out each author participates in which topics. For example, Writer A attends seminar with topic Science and Children. He also writes a book with topic Science and Arts. So, we could say that he participates in Science, Children and Arts.

(Q5)

```
Run SQL
1 WITH evt_info AS (SELECT a.evt_id, b.top_name FROM ydzc_evt a,
ydzc_top b, ydzc_evt_top c WHERE a.evt_id=c.evt_id AND b.
top_name=c.top_name) SELECT DISTINCT c.cust_id Customer_ID, a.
top_name Topic FROM evt_info a, ydzc_exh b, ydzc_cust c,
ydzc_reg d WHERE a.evt_id=b.evt_id AND b.evt_id=d.evt_id AND c.
cust_id=d.cust_id ORDER BY Customer_ID;
2
```

			* Customer_ID 	* Topic 
			Filter	Filter
	1		1	Arts
	2		1	History
	3		2	Science
	4		2	Arts
	5		2	Children
	6		2	History
	7		3	Arts
	8		3	History
	9		5	Travel
	10		5	Arts
	11		5	History
	12		6	Arts
	13		6	Travel
	14		7	Arts
	15		7	History
	16		8	Arts
	17		8	History
	18		9	Children
	19		9	Science

Explanation: We want to find out exhibition of which kind attract each customer. For example, customer (ID=1) has registered exhibition(s) of both Arts and History topics. (We could use customer name here, but it might be possible that two customer owns the same name)

(Q6)

```
Run SQL
1 WITH fund_rank AS (SELECT a.evt_name Event_Name, b.
  evtsem_totfund Fund, RANK() OVER (ORDER BY b.evtsem_totfund
  DESC) myrank FROM ydzc_evt a, ydzc_sem b WHERE a.evt_id=b.
  evt_id) SELECT Event_Name, Fund FROM fund_rank WHERE myrank<=4;
2
```

<input checked="" type="checkbox"/>	<input type="checkbox"/>	* Event_Name	* Fund
		Filter	Filter
	1	Crystal Mechanics Future	2000.00
	2	Quantum Mechanics Futu	1500.00
	3	Flow Mechanics Future	600.00
	4	Structural Mechanics Futu	600.00

Explanation: We want to show top 4 (N=4) seminars with highest fund.