

# Project

## Order Processing System

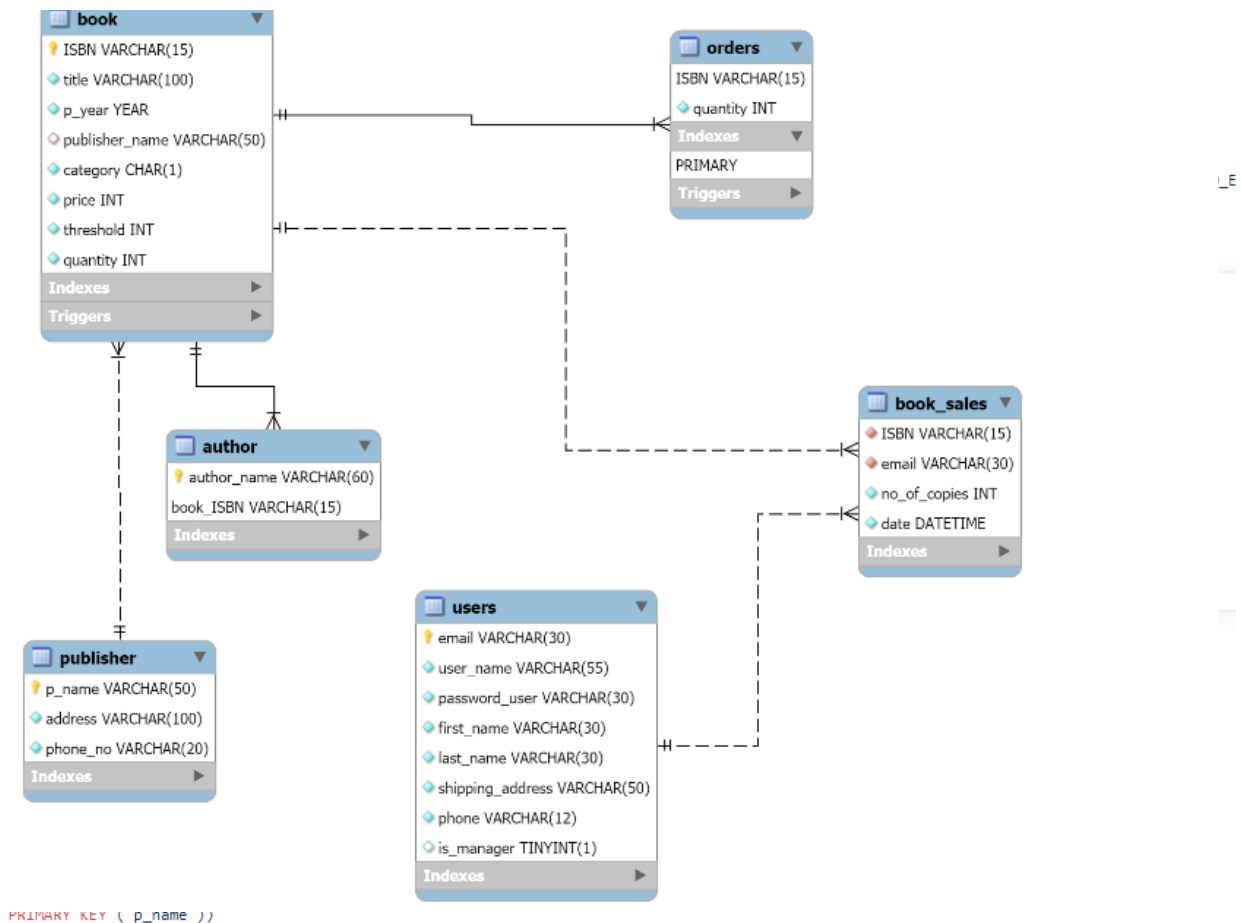
### Group Members:

Toka Alaa Ahmed (16)

Abdelmonem Elsayy (28)

Nada Salama Mohammed (58)

## ERD Diagram:



- ***We create first schema called bookstore***
- ***Then create inside it 6 tables : book, author, publisher, users, book\_sales***
- ***& 1 PROCEDURE : order\_insertion***
- ***& 5 function : add\_book, log\_in, order\_book, sign\_up, update\_user\_info***

```

32     PRIMARY KEY (`p_name`))
33 ENGINE = InnoDB;
34
35
36 -----
37 -- Table `Bookstore`.`Book`
38 -----
39 CREATE TABLE IF NOT EXISTS `Bookstore`.`Book` (
40     `ISBN` VARCHAR(15) NOT NULL,
41     `title` VARCHAR(100) NOT NULL,
42     `p_year` YEAR NOT NULL,
43     `publisher_name` VARCHAR(50) NULL,
44     `category` CHAR(1) NOT NULL,
45     `price` INT NOT NULL,
46     `threshold` INT NOT NULL,
47     `quantity` INT NOT NULL,
48     PRIMARY KEY (`ISBN`),
49     UNIQUE INDEX `title_UNIQUE` (`title` ASC) VISIBLE,
50     INDEX `p_name_idx` (`publisher_name` ASC) VISIBLE,
51     CONSTRAINT `p_name`
52         FOREIGN KEY (`publisher_name`)
53         REFERENCES `Bookstore`.`Publisher` (`p_name`)
54         ON DELETE SET NULL
55         ON UPDATE CASCADE)
56 ENGINE = InnoDB;

```

```

59  -- -----
60  -- Table `Bookstore`.`Author`
61  -- -----
62  CREATE TABLE IF NOT EXISTS `Bookstore`.`Author` (
63      `author_name` VARCHAR(60) NOT NULL,
64      `book_ISBN` VARCHAR(15) NOT NULL,
65      PRIMARY KEY (`author_name`, `book_ISBN`),
66      INDEX `author_isbn_idx` (`book_ISBN` ASC) VISIBLE,
67      CONSTRAINT `author_isbn`
68          FOREIGN KEY (`book_ISBN`)
69          REFERENCES `Bookstore`.`Book` (`ISBN`)
70          ON DELETE CASCADE
71          ON UPDATE CASCADE)
72  ENGINE = InnoDB;
73
74
75  -- -----
76  -- Table `Bookstore`.`users`
77  -- -----
78  CREATE TABLE IF NOT EXISTS `Bookstore`.`users` (
79      `email` VARCHAR(30) NOT NULL,
80      `user_name` VARCHAR(55) NOT NULL,
81      `password_user` VARCHAR(30) NOT NULL,
82      `first_name` VARCHAR(30) NOT NULL,
83      `last_name` VARCHAR(30) NOT NULL,
84      `shipping_address` VARCHAR(50) NOT NULL,
85      `phone` VARCHAR(12) NOT NULL,
86      `is_manager` TINYINT(1) NULL,
87      UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE,
88      PRIMARY KEY (`email`),
89      UNIQUE INDEX `user name UNIQUE` (`user name` ASC) VISIBLE)

```

---

---

```
93  -- -----
94  -- Table `Bookstore`.`Book_sales`
95  -- -----
96  CREATE TABLE IF NOT EXISTS `Bookstore`.`Book_sales` (
97    `ISBN` VARCHAR(15) NOT NULL,
98    `email` VARCHAR(30) NOT NULL,
99    `no_of_copies` INT NOT NULL,
100   `date` DATETIME NOT NULL,
101   `totalprice` INT,
102   INDEX `sales_isbn_idx` (`ISBN` ASC) VISIBLE,
103   INDEX `sales_email_idx` (`email` ASC) VISIBLE,
104   CONSTRAINT `sales_isbn`
105     FOREIGN KEY (`ISBN`)
106     REFERENCES `Bookstore`.`Book` (`ISBN`)
107     ON DELETE CASCADE
108     ON UPDATE CASCADE,
109   CONSTRAINT `sales_email`
110     FOREIGN KEY (`email`)
111     REFERENCES `Bookstore`.`users` (`email`)
112     ON DELETE CASCADE
113     ON UPDATE CASCADE)
114  ENGINE = InnoDB;
115
116
117  -- -----
118  -- Table `Bookstore`.`orders`
119  -- -----
120  CREATE TABLE IF NOT EXISTS `Bookstore`.`orders` (
121    `ISBN` VARCHAR(15) NOT NULL,
122    `quantity` INT NOT NULL,
123    PRIMARY KEY (`ISBN`),
```

```

124     CONSTRAINT `order_isbn2`
125     FOREIGN KEY (`ISBN`)
126     REFERENCES `Bookstore`.`Book` (`ISBN`)
127     ON DELETE CASCADE
128     ON UPDATE CASCADE)
129 ENGINE = InnoDB;
130
131 USE `bookstore` ;
132
133 -----
134 -- Table `bookstore`.`publisher`
135 -----
136 CREATE TABLE IF NOT EXISTS `bookstore`.`publisher` (
137   `p_name` VARCHAR(50) NOT NULL,
138   `address` VARCHAR(100) NOT NULL,
139   `phone_no` VARCHAR(20) NOT NULL,
140   PRIMARY KEY (`p_name`))
141 ENGINE = InnoDB
142 DEFAULT CHARACTER SET = utf8;
143
144
145 -----
146 -- Table `bookstore`.`book`
147 -----
148 CREATE TABLE IF NOT EXISTS `bookstore`.`book` (
149   `ISBN` VARCHAR(15) NOT NULL,
150   `title` VARCHAR(100) NOT NULL,
  
```

```

151     `p_year` YEAR NOT NULL,
152     `publisher_name` VARCHAR(50) NULL DEFAULT NULL,
153     `category` CHAR(1) NOT NULL,
154     `price` INT NOT NULL,
155     `threshold` INT NOT NULL,
156     `quantity` INT NOT NULL,
157     PRIMARY KEY (`ISBN`),
158     UNIQUE INDEX `title_UNIQUE` (`title` ASC) VISIBLE,
159     INDEX `p_name_idx` (`publisher_name` ASC) VISIBLE,
160     CONSTRAINT `p_name`
161         FOREIGN KEY (`publisher_name`)
162         REFERENCES `bookstore`.`publisher` (`p_name`)
163         ON DELETE SET NULL
164         ON UPDATE CASCADE)
165 ENGINE = InnoDB
166 DEFAULT CHARACTER SET = utf8;
167
168
169 -- -----
170 -- Table `bookstore`.`author`
171 -- -----
172 CREATE TABLE IF NOT EXISTS `bookstore`.`author` (
173     `author_name` VARCHAR(60) NOT NULL,
174     `book_ISBN` VARCHAR(15) NOT NULL,
175     PRIMARY KEY (`author_name`, `book_ISBN`),
176     INDEX `author_isbn_idx` (`book_ISBN` ASC) VISIBLE,
177     CONSTRAINT `author_isbn`
178         FOREIGN KEY (`book_ISBN`)
179         REFERENCES `bookstore`.`book` (`ISBN`)

```

---

```

180         ON DELETE CASCADE
181         ON UPDATE CASCADE)
182 ENGINE = InnoDB
183 DEFAULT CHARACTER SET = utf8;
184
185
186 -----
187 -- Table `bookstore`.`users`
188 -----
189 CREATE TABLE IF NOT EXISTS `bookstore`.`users` (
190     `email` VARCHAR(30) NOT NULL,
191     `user_name` VARCHAR(55) NOT NULL,
192     `password_user` VARCHAR(30) NOT NULL,
193     `first_name` VARCHAR(30) NOT NULL,
194     `last_name` VARCHAR(30) NOT NULL,
195     `shipping_address` VARCHAR(50) NOT NULL,
196     `phone` VARCHAR(12) NOT NULL,
197     `is_manager` TINYINT(1) NULL DEFAULT NULL,
198     PRIMARY KEY (`email`),
199     UNIQUE INDEX `email_UNIQUE` (`email` ASC) VISIBLE,
200     UNIQUE INDEX `user_name_UNIQUE` (`user_name` ASC) VISIBLE)
201 ENGINE = InnoDB
202 DEFAULT CHARACTER SET = utf8;
203
204
205 -----
206 -- Table `bookstore`.`book_sales`
207 -----
208 CREATE TABLE IF NOT EXISTS `bookstore`.`book_sales` (
209     `ISBN` VARCHAR(15) NOT NULL,
210     `email` VARCHAR(30) NOT NULL,

```



```

210     `email` VARCHAR(30) NOT NULL,
211     `no_of_copies` INT NOT NULL,
212     `sell_date` DATETIME NOT NULL,
213     INDEX `sales_isbn_idx` (`ISBN` ASC) VISIBLE,
214     INDEX `sales_email_idx` (`email` ASC) VISIBLE,
215     CONSTRAINT `sales_email`
216         FOREIGN KEY (`email`)
217         REFERENCES `bookstore`.`users` (`email`)
218         ON DELETE CASCADE
219         ON UPDATE CASCADE,
220     CONSTRAINT `sales_isbn`
221         FOREIGN KEY (`ISBN`)
222         REFERENCES `bookstore`.`book` (`ISBN`)
223         ON DELETE CASCADE
224         ON UPDATE CASCADE)
225     ENGINE = InnoDB
226     DEFAULT CHARACTER SET = utf8;
227
228     USE `bookstore` ;
229

```

```

CREATE DEFINER=`root`@`localhost` FUNCTION `log_in`(email_user varchar(30), pass_user varchar(30))
    DETERMINISTIC
) BEGIN
    declare msg varchar(500);
    if (SELECT EXISTS(SELECT * from users WHERE email = email_user) = 0)
) then set msg = concat('Please check your email, you should register before.....');
    elseif (! (select(SELECT password_user from users WHERE email = email_user) LIKE pass_user))
    then set msg = concat('password is incorrect, Please check your password.....');
    else
    set msg = concat('You log_in Successfully.....');
    end if ;
    return msg ;
End

```

***We use this function to allow user to log\_in with his account where it is created before and we do our check such that ensure that this email exist or not and check the correctness of password.***

```

-- -----
-- function order_book
-- -----

DELIMITER $$
USE `bookstore`$$
CREATE DEFINER=`root`@`localhost` FUNCTION `order_book`(ordered_ISBN varchar(15), copies_no int, user_email varchar(30),
sell_date DATE) RETURNS varchar(500) CHARSET utf8
    DETERMINISTIC
BEGIN
declare msg varchar(500);
declare num int;
SELECT quantity into num FROM book
WHERE book.ISBN = ordered_ISBN;
IF copies_no <= num THEN set msg = concat('The order is done correctly.');
```

*We use this function to check if the order will do correct or not where we check on copies\_num is less than or equal quantity.*

```

ELSE set msg = concat("The number of books can't be negative");
END IF;
RETURN msg;
END$$

DELIMITER ;

```

```

303 -- -----
304 -- function sign_up
305 -- -----
306
307 DELIMITER $$
308 USE `bookstore`$$
309 CREATE DEFINER=`root`@`localhost` FUNCTION `sign_up`( email_user varchar(30) ,
310     user_name varchar(55) ,
311     password_user varchar(30) ,
312     first_name varchar(30) ,
313     last_name varchar(30) ,
314     shipping_address varchar(50) ,
315     phone varchar(12)) RETURNS varchar(500) CHARSET utf8
316     DETERMINISTIC
317 BEGIN
318 declare msg varchar(500);
319 if (SELECT EXISTS(SELECT * from users WHERE email = email_user) = 1)
320 then set msg = concat('Please check your email, this email exists already.....');
```

*We use this function to allow user create account and then add it to table*

```

321 else
322 set msg = concat('You sign_up Successfully.....');
```

*We use this function to allow user create account and then add it to table*

```

323 INSERT INTO users VALUES(email_user , user_name , password_user , first_name ,
324     last_name , shipping_address , phone, false );
325 end if ;
326 return msg ;
327 End$$
328
329 DELIMITER ;
330 -- -----

```

```

352 -- -----
353 -- function update_user_info
354 -- -----
355
356 DELIMITER $$
357 USE `bookstore`$$
358 CREATE DEFINER=`root`@`localhost` FUNCTION `update_user_info`(user_email varchar(30) ,
359     new_user_name varchar(55), new_password_user varchar(30) ,
360     new_first_name varchar(30) , new_last_name varchar(30) ,
361     new_shipping_address varchar(50) , new_phone varchar(12)) RETURNS tinyint(1)
362     DETERMINISTIC
363 BEGIN
364     declare done boolean DEFAULT true;
365     UPDATE users
366     SET user_name = new_user_name ,
367         password_user = new_password_user,
368         first_name = new_first_name,
369         last_name = new_last_name,
370         shipping_address = new_shipping_address,
371         phone = new_phone
372     WHERE email = user_email;
373     return done ;
374 End$$
375
376 DELIMITER ;
377 USE `Bookstore`;
378
379 DELIMITER $$
380 USE `bookstore`$$
381 CREATE

```

***This function will be called if user want change his information***

```

329 DELIMITER ;
330 -- -----
331 -- procedure order_insertion
332 -- -----
333
334 USE `bookstore`;
335 DROP procedure IF EXISTS `order_insertion`;
336
337 DELIMITER $$
338 USE `bookstore`$$
339 CREATE DEFINER=`root`@`localhost` PROCEDURE `order_insertion`(IN ordered_ISBN varchar(15),IN copies_no int,IN user_email varchar(30),
340     IN sell_date DATETIME)
341 BEGIN
342     declare book_price int;
343     SELECT price into book_price FROM book
344     WHERE ordered_ISBN = book.ISBN;
345     UPDATE book SET book.quantity = book.quantity - copies_no
346     WHERE book.ISBN = ordered_ISBN;
347     INSERT INTO book_sales VALUES(ordered_ISBN, user_email, copies_no, sell_date, book_price * copies_no);
348 END$$
349
350 DELIMITER ;
351
352 -- -----
353 -- function update_user_info
354 -- -----
355
356 DELIMITER $$
357 USE `bookstore`$$
358 CREATE DEFINER=`root`@`localhost` FUNCTION `update_user_info`(user_email varchar(30) ,
359     new_user_name varchar(55), new_password_user varchar(30) ,

```

***this procedure will be called to order which add this order to table and it is called after checking of copies\_no***

```
381 CREATE
382 DEFINER=`root`@`localhost`
383 TRIGGER `bookstore`.`confirm_order`
384 BEFORE DELETE ON `bookstore`.`orders`
385 FOR EACH ROW
386 BEGIN
387     UPDATE book
388     SET book.quantity = book.quantity + OLD.quantity
389     WHERE book.ISBN = OLD.ISBN;
390 END$$
391
392
393 DELIMITER ;
394 USE `bookstore`;
395
396 DELIMITER $$
397 USE `bookstore`$$
398 CREATE
399 DEFINER=`root`@`localhost`
400 TRIGGER `bookstore`.`modify_existing_books`
401 BEFORE UPDATE ON `bookstore`.`book`
402 FOR EACH ROW
403 BEGIN
404     IF NEW.quantity < 0 THEN
405         SIGNAL SQLSTATE '45000'
406         SET MESSAGE_TEXT = "The number of books can't be negative";
407     END IF;
408 END$$
409
410 USE `bookstore`$$
411 CREATE
```

---

```

411 CREATE
412 DEFINER='root'@'localhost'
413 TRIGGER `bookstore`.`orders_on_books`
414 AFTER UPDATE ON `bookstore`.`book`
415 FOR EACH ROW
416 BEGIN
417 IF NEW.quantity < NEW.threshold THEN
418     if (SELECT EXISTS(SELECT * from orders WHERE orders.ISBN = new.ISBN) = 0) then
419         INSERT INTO orders VALUES(NEW.ISBN,2 * NEW.threshold);
420     end if;
421
422 END IF;
423 END$$
424
425
426 DELIMITER ;
427
428 SET SQL_MODE=@OLD_SQL_MODE;
429 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
430 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

---

## Program Screenshots:

### Customer:



**Welcome**

**email :**

**password :**

**LOG\_IN**

**SIGN\_UP**

Sign\_up

*Create new Account*

Email : toka.alaa

Password : 1111

User\_Name : tokaalaa

First\_Name : toka

Last\_Name : alaa

Phone : 444455555

Shipping\_Address : alex

**CREATE**

Log\_in


**Welcome**

email : toka.alaa

password : 1111

**LOG\_IN**

**SIGN\_UP**

 Home

[LOG\\_OUT](#) [PROFILE](#) [CART](#) [HOME](#)

Select attribute  
which you will  
search by:


☐ ISBN

☐ Title

☐ Author

☐ Publisher

☐ Specific  
Category

 Home

[LOG\\_OUT](#) [PROFILE](#) [CART](#) [HOME](#)

Select attribute  
which you will  
search by:

☒ ISBN

☐ Title

☐ Author

☐ Publisher

☐ Specific  
Category

Searched key :

[SEARCH](#)



Search



**LOG\_OUT**

**PROFILE**

**CART**

**HOME**

**Book\_ID**   **Book\_Title**   **Book\_Price**   **Page number 1 of total 1**

Previous

Next

nada3

nada3

50

1



Add to Cart

Info. of book



Book information

Information of selected book

ISBN :

nada3

Title :

nada3

Publisher year :

2008-01-01

Publisher name :

nada

Category :

s

Price :

50

Available quantity :

5

OK

Cart

LOG\_OUT

PROFILE

CART

HOME

Book_ID	Book_Title	Order_Price	Page number 1 of total 1
nada3	nada3	1 * 50 = 50	<div>Previous</div> <div>Next</div> <div>Remove</div> <div>Info. of book</div>

Total\_Price :

50

CONFIRM\_ALL

OR

REMOVE\_ALL

Credit\_card Frame

**Credit card number :** \*(Require 16 digit)

1234567890

**Expiry date :**

May 2020

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

today: May 22, 2020

Credit\_card Frame

**Credit card number :** \*(Require 16 digit)

1234567890

**Expiry date :** Jun 30, 2020

**Order ^\_^**

Update

LOG\_OUT

PROFILE

CART

HOME

Edit my personal information

Email :

toka.alaa

Password :

1111

User\_Name :

tokaalaa

First\_Name :

toka

Last\_Name :

alaa

Phone :

444455555

Shipping\_Address :

alex

UPDATE

Manager:

Log\_in

Welcome

email :

nada2

password :

123

LOG\_IN

SIGN\_UP

Home

LOG\_OUT

PROFILE

CART

HOME

MANAGE

Select attribute  
which you will  
search by:

☐ ISBN

☐ Title

☐ Author

☐ Publisher

☐ Specific  
Category

Manage Frame

LOG\_OUT

PROFILE

CART


HOME

MANAGE

Manage Database

Promote Users

Manage Sales



—


□

×

☐ Add book

☐ Confirm order

☐ Order book



—

□

×

book

ISBN

12341234

Title

Arch

Category

s

price

30

Threshold

20

Quantity

30

Publication year

2008

ADD

publisher

Name

nour

Address

alex

Telephone

44444444

Authors

ahmed  
hamed

Delete

Add name

Library Management System

☐ Add book  
☐ Confirm order  
☒ Order book

ISBN: 12341234      Quantity: 10

**Order**

—□×

☐ Add book

☒ Confirm order

☐ Order book

ISBN

12341234

Confirm

Manage Frame

—□×

LOG\_OUT

PROFILE

CART

HOME

MANAGE

Manage Database

Promote Users

—□×

User's Email

toka.alaa

Promote





- ☒ The total sales for books in the previous month
- ☐ The top 5 customers for the last three months
- ☐ The top 10 selling books for the last three months

ISBN	Total Sales
nada = 1	
nada10 = 2	
nada12 = 1	
nada3 = 1	
nada5 = 3	