

BIL372 DATABASE SYSTEMS

## TERM PROJECT REPORT

### DESIGN AND IMPLEMENTATION OF BOOK ORDER SYSTEM

Baran Berk BACALAN

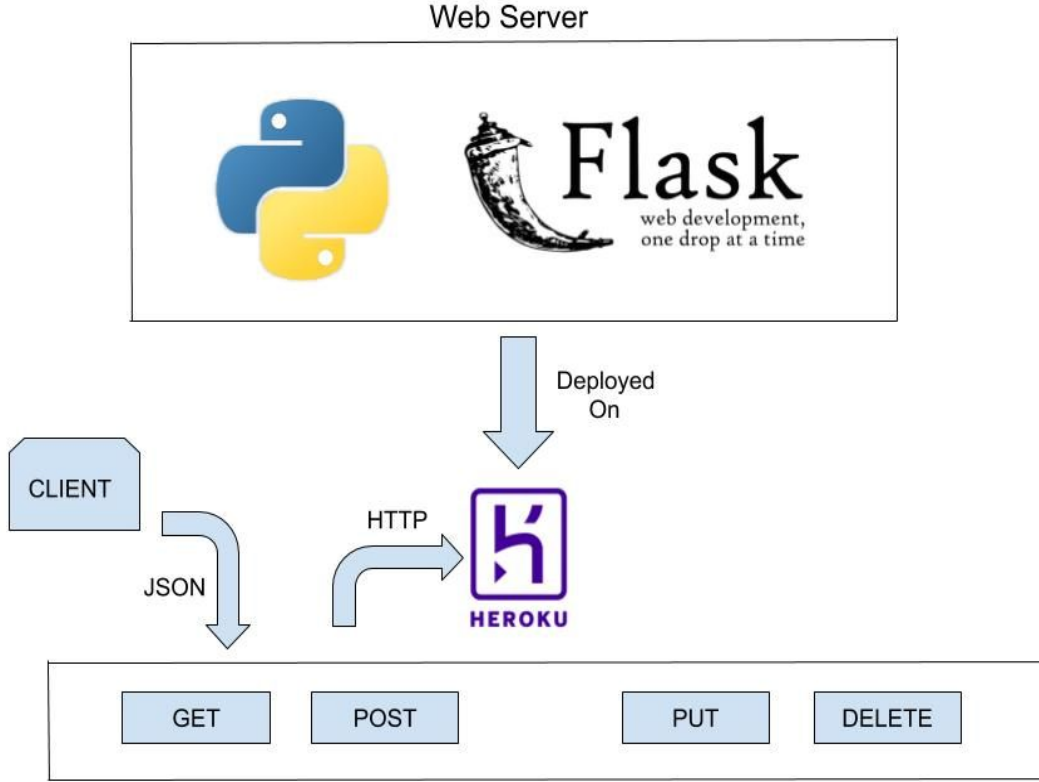
Ege TAN

Erhan Baturay ONURAL

Erhan EKER



# 1.Introduction



2019-2020 Güz dönemi Bil372 dersinde dönem projesi olarak kitap sipariş sistemi yaptık. Yaptığımız projede kullanıcıların “sipariş verme” arayüzünde sepetlerine kitap ekleme/çıkarma işlemlerini yapabilmeleri sağlanmıştır. Devam eden sipariş verme sürecinde alacakları kitapların toplam tutarını ve kaç adet aldıklarını görebildikleri bir arayüzle karşılaşan kullanıcılar, siparişlerini onaylayarak siparişlerini tamamlayabiliyorlar. Ayrıca Kullanıcılar tarafından tutulan bilgiler sayesinde kullanıcılar dilerse son siparişlerini görüntüleyebilir ya da profilindeki adres, kullanıcı adı, telefon numarası gibi özellikleri düzenleyebilirler. Aynı şekilde admin paneli aracılığıyla yeni kitaplar

oluřturulabilir. Bu kitaplar daha sonra yine aynı arayüz yardımıyla güncellenebilir veya silinmek istendiğı durumda silinebilir. BookIt adını verdiğimiz kitap satış platformumuz birden çok kullanıcı kullanımını aynı anda desteklemektedir. Her bir kullanıcı için ayrı bir kayıt bilgisi tutar ve bunu kendi içerisinde yönetir.

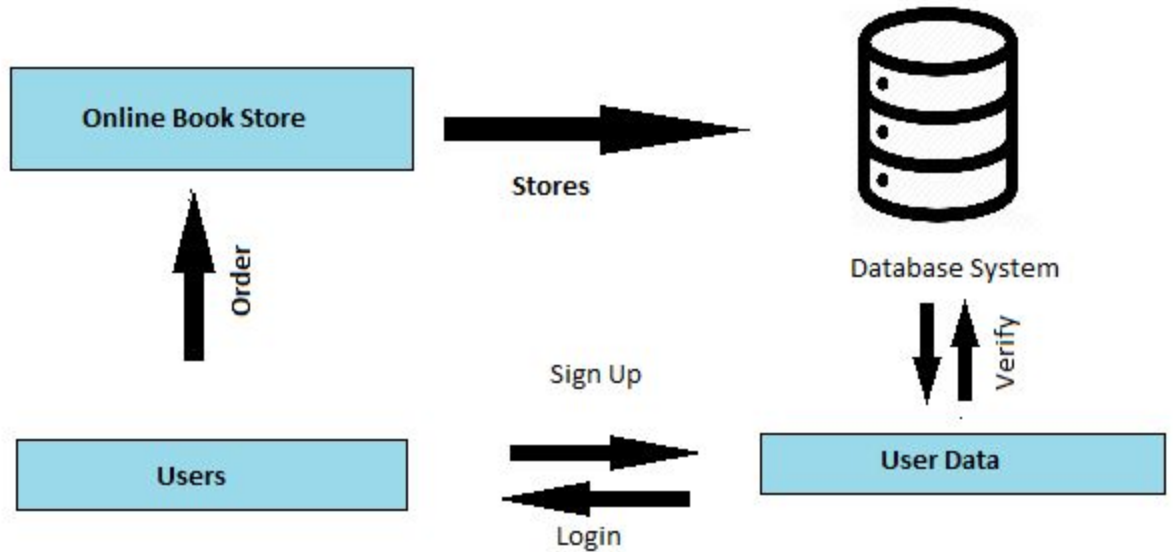
**Gitstat Dosyalarımız:**

[https://drive.google.com/drive/folders/1\\_Z\\_mh7qsWxHTMGpYAKZBvYWLva2qSHKk?usp=sharing](https://drive.google.com/drive/folders/1_Z_mh7qsWxHTMGpYAKZBvYWLva2qSHKk?usp=sharing)

**Github Web Repomuz:**

<https://github.com/berk-12/BookStore>

## 2.HIPO



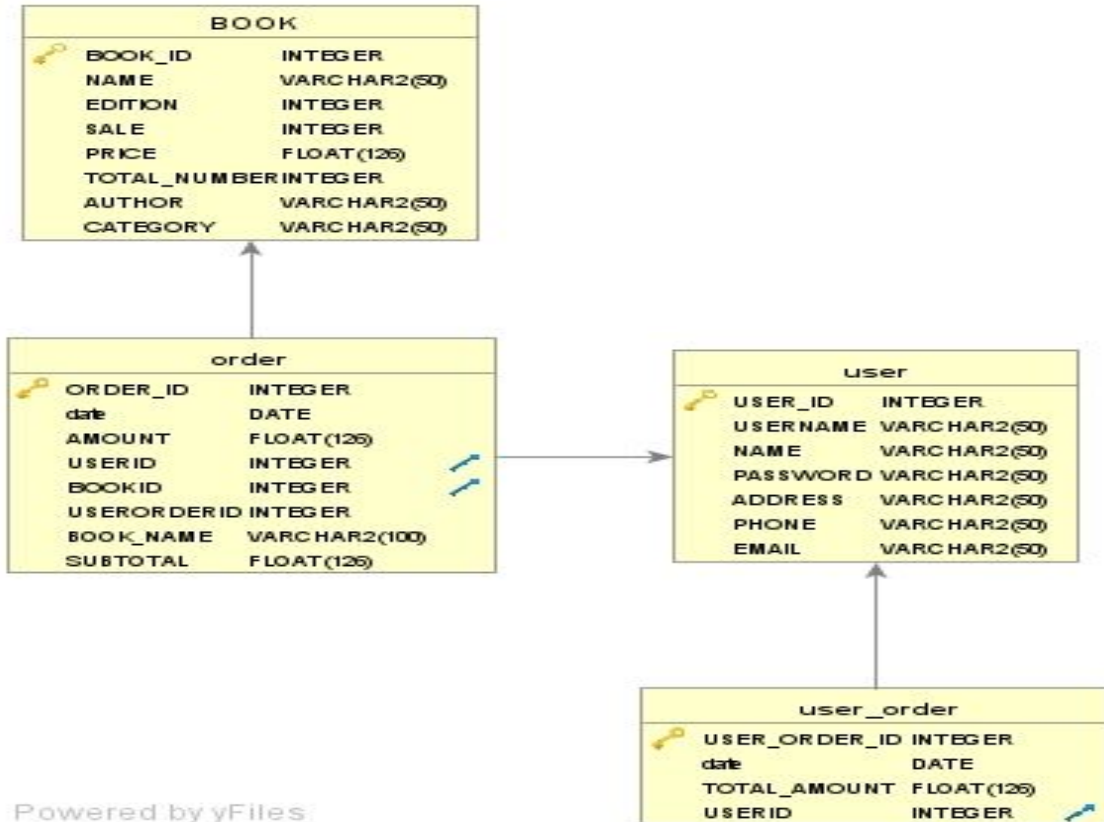
Müşteri öncelikle sisteme kayıt olur veya giriş yapar.Adminin eklemiş olduğı kitap listesinden sepetine kitap ekler ve sipariş eder.Müşteri kendisi bilgilerini düzenleyebilir.Adresini mailini telefon numarasını ve adını değıřtirebilir.Önceden

vermiş olduğu siparişleri kontrol edebilir. Sepetine eklediği kitapların toplam ücretini görebilir ve finish diyerek siparişi verebilir.

Müşteri kitaplar arasından search ederek kitapları kitabın ismine, yazarın adına ve kitabın türüne göre filtreleyerek görüntüleyebilir..

Admin ise kitap ekleyebilir kitapların ücretini adını kategorisini ekleyebilir, düzenleyebilir veya silebilir. Her userın (müşterinin) kendi hesabındaki bilgiler ve verdiği siparişler ayrı ayrı tutulur. Kullanıcı bunları Kendi sayfasında görüntüleyebilir.

### 3.DB Schema



## 4.Functions

### 4.1 Login:



Please sign in

<input type="text"/>
<input type="password"/>
<input type="button" value="Sign In"/>
<input type="button" value="Not an user, sign up!"/>

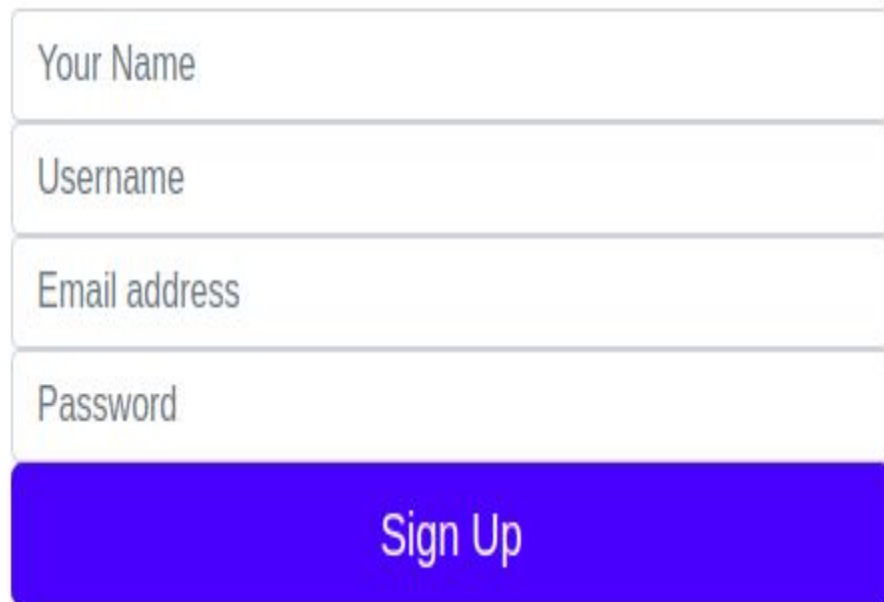
SQL:

```
SELECT PASSWORD  
FROM USER  
WHERE USERNAME = USER.USERNAME
```

→ ELDE EDİLEN PAROLA KONTROL EDİLİR.

## 4.2 Register:

Please sign up



A registration form with four input fields and a submit button. The fields are labeled 'Your Name', 'Username', 'Email address', and 'Password'. Below the fields is a blue button with the text 'Sign Up'.

**SQL:**

```
INSERT INTO USER
  (USERNAME, NAME, PASSWORD, EMAIL)
(FORM.USERNAME, FORM.YOUR_NAME, FORM.PASSWORD,
 FORM.EMAIL)
```

→ GİRİLEN BİLGİLERLE KULLANICI YARATILIR.

### 4.3 Add Book(Admin):



The image shows a web form titled "Create Books" on a light blue background. The form contains seven input fields, each with a label to its left: "Name", "Author", "Edition", "Category", "Sale", "Price", and "Total Number". Below these fields is a button labeled "Create Book".

#### SQL:

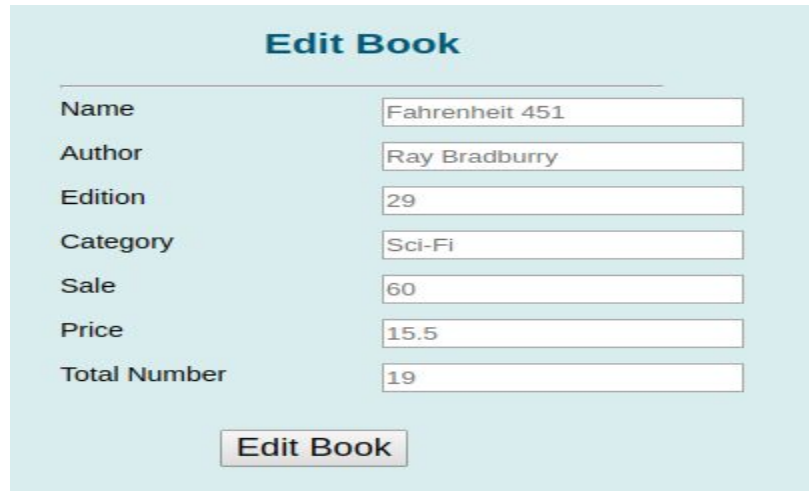
INSERT INTO BOOK

(NAME, EDITION, SALE, PRICE, TOTAL\_NUMBER, AUTHOR,  
CATEGORY)

(FORM.NAME, FORM.EDITION, FORM.SALE, FORM.PRICE,  
FORM.TOTAL\_NUMBER, FORM.AUTHOR, FORM.CATEGORY)

→ GİRİLEN BİLGİLERLE KİTAP YARATILIR.

## 4.4 Edit Book(Admin):



Name	Fahrenheit 451
Author	Ray Bradbury
Edition	29
Category	Sci-Fi
Sale	60
Price	15.5
Total Number	19

Edit Book

### SQL:

#### UPDATE BOOK

SET NAME = FORM.NAME, AUTHOR=FORM.AUTHOR, EDITION =  
FORM.EDITION, CATEGORY=FORM.CATEGORY, SALE=FORM.SALE,  
PRICE=FORM.PRICE, TOTAL\_NUMBER=FORM.TOTAL\_NUMBER  
WHERE BOOK\_ID = FORM.BOOK\_ID

→ GİRİLEN BİLGİLERLE KİTAP GÜNCELLENİR.

→ BURADA HANGİ KİTABIN GÜNCELLENECEĞİ BİLGİSİNİ KİTAP ID  
BİLGİSİ ÜSTÜNDEN SAĞLIYORUZ.



## 4.5 Delete Book(Admin):

### BookIt - Admin View

#### Books

Book Name	Author	Edition	Category	Sale	Total Number	Price	Actions
Fahrenheit 451	Ray Bradbury	29	Sci-Fi	60	19	15.5	<a href="#">Delete</a> <a href="#">Edit</a>
Dubrovski	Sergeyevic Puskin	27	Russian Classics	39	439	12.75	<a href="#">Delete</a> <a href="#">Edit</a>
Me	Elton John	3	Autobiography	21	116	21.5	<a href="#">Delete</a> <a href="#">Edit</a>

#### SQL:

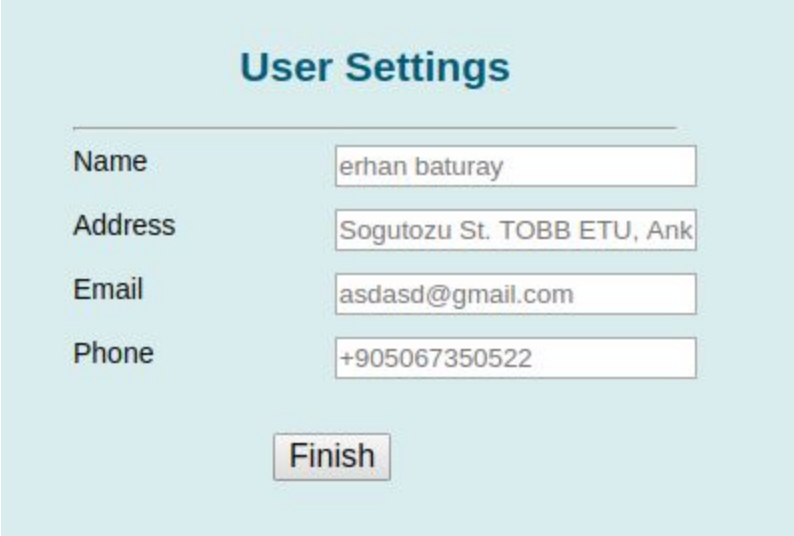
DELETE FROM BOOK

WHERE BOOK\_ID = FORM.BOOK\_ID

→ İSTENEN KİTAP TABLODAN SİLİNİR.

→ BURADA HANGİ KİTAPIN SİLİNECEĞİ BİLGİSİNİ KİTAP ID BİLGİSİ ÜSTÜNDEN SAĞLIYORUZ.

## 4.6 User Settings:



The image shows a web form titled "User Settings" on a light blue background. The form contains four input fields with labels to their left: "Name" with the value "erhan baturay", "Address" with "Sogutozu St. TOBB ETU, Ank", "Email" with "asdasd@gmail.com", and "Phone" with "+905067350522". Below these fields is a "Finish" button.

User Settings	
Name	erhan baturay
Address	Sogutozu St. TOBB ETU, Ank
Email	asdasd@gmail.com
Phone	+905067350522
<input type="button" value="Finish"/>	

### SQL:

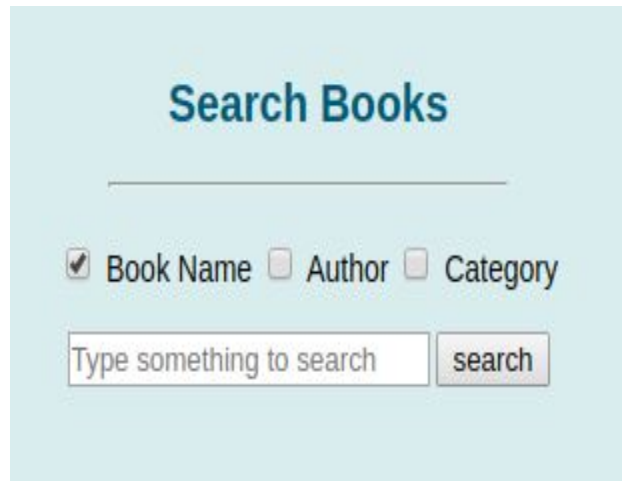
UPDATE USER

SET NAME = FORM.NAME, ADDRESS=FORM.ADDRESS, EMAIL =  
FORM.EMAIL, PHONE=FORM.PHONE  
WHERE USER\_ID = FORM.USER\_ID

→ İSTENEN KULLANICININ BİLGİSİ GÜNCELLENİR.

→ BURADA HANGİ KULLANICININ BİLGİSİNİN GÜNCELLENECEĞİNİ  
USER ID BİLGİSİ ÜSTÜNDEN SAĞLIYORUZ.

## 4.7 SearchBooks:



The image shows a web form titled "Search Books" in a light blue box. Below the title is a horizontal line. Underneath the line are three radio buttons: "Book Name" (which is checked), "Author", and "Category". Below these radio buttons is a text input field with the placeholder text "Type something to search" and a "search" button to its right.

### SQL:

```
SELECT NAME ( + OTHER SELECTED OPTIONS)
FROM BOOK
WHERE NAME ( + OTHER SELECTED OPTIONS)
LIKE FORM.SEARCHTEXT
```

→ GİRİLEN GİRDİ SEÇİLİ OPSİYONLAR ARASINDA TABLO  
SÜTUNLARINDA ARANIR.

→ BİRDEN ÇOK İŞARETLEME YAPILMASI DURUMUNDA SORGU  
SONUÇLARI BİRLEŞTİRİLİR.

## 4.8 Add to cart:

Books							
Book Name	Author	Edition	Category	Sale	Total Number	Price	Actions
Fahrenheit 451	Ray Bradbury	29	Sci-Fi	60	19	15.5	<a href="#">Add to cart</a>
Dubrovski	Sergeyevic Puskin	27	Russian Classics	39	439	12.75	<a href="#">Add to cart</a>
Me	Elton John	3	Autobiography	21	116	21.5	<a href="#">Add to cart</a>

## Sepete ekledikten sonra sepet görünümü:

Shopping Cart					
Book Name	Author	Edition	Price	Amount	Actions
Fahrenheit 451	Ray Bradbury	29	15.5	1	<a href="#">Remove</a>
Dubrovski	Sergeyevic Puskin	27	12.75	3	<a href="#">Remove</a>
Me	Elton John	3	21.5	1	<a href="#">Remove</a>

Complete Order

**SQL: (kitap listesini göstermek için)**

```
SELECT NAME AUTHOR EDITION PRICE AMOUNT  
FROM BOOK  
WHERE BOOKID = FORM.BOOKID
```

→ BOOKID İLE ULAŞILAN KİTAP BİLGİSİ TABLODA GÖSTERİLİR.

→ ADD TO CART BUTONU İLE SEPETE TAŞINIR.

## 4.9 Payment:

### Payment Info

---

Name	<input type="text" value="erhan baturay"/>
Address	<input type="text" value="Sogutozu St. TOBB ETU, Ank"/>
Email	<input type="text" value="asdasd@gmail.com"/>
Phone	<input type="text" value="+905067350522"/>

Total: 47.0Finish

### Payment Done!

---

Your order has been added to orders list.

Back

### SQL: (ORDER OLUŞTURMA)

```
INSERT INTO ORDER
(DATE, AMOUNT, USERID, BOOKID, USER_ORDERID, BOOKNAME,
SUBTOTAL)
(CURRENTDATE(), FORM.AMOUNT, FORM.USERID, FORM.BOOKID,
FORM.USER_ORDERID, FORM.BOOKNAME, FORM.SUBTOTAL)
```

### SQL: (USER\_ORDER OLUŞTURMA)

```
INSERT INTO USER_ORDER
(DATE, TOTAL_AMOUNT, USERID)
(CURRENTDATE(), FORM.AMOUNT, FORM.USERID)
```

→ SEPETTE OLAN KİTAPLARIN HER BİRİ İÇİN 1 ADET ORDER SATIRI TABLOYA EKLENİR.

→ EKLENEN HER BİR SATIRIN USER\_ORDERID AYNI SİPARİŞLER İÇİNDE YER ALAN ORDERLAR İÇİN AYNIDIR. BU SÜTUN SATIN ALIMLARIN HANGİ SİPARİŞ İÇİNDE YER ALDIĞI BİLGİSİNİ TUTAR.

## 4.10 Recent Orders:

### Recent Orders

Date	Book Name	Amount	Subtotal
2019-11-25	Fahrenheit 451	3.0	47.25
2019-11-25	Dubrovski	1.0	12.75
2019-11-25	Me	1.0	21.5
2019-11-25	Fahrenheit 451	1.0	15.75
2019-11-25	Dubrovski	1.0	12.75
2019-11-25	Me	1.0	21.5
2019-11-25	Me	1.0	21.5
2019-11-25	Dubrovski	1.0	12.75
2019-11-25	Dubrovski	2.0	25.5
2019-11-26	Fahrenheit 451	1.0	15.5
2019-11-26	Dubrovski	3.0	38.25
2019-11-26	Me	1.0	21.5
2019-11-26	Fahrenheit 451	2.0	31.0
2019-11-26	Dubrovski	2.0	25.5
2019-11-26	Me	2.0	43.0
2019-11-26	Dubrovski	2.0	25.5
2019-11-26	Me	1.0	21.5

## SQL:

```
SELECT order.date,book.bookname,order.amount,order.subtotal from  
order INNER JOIN book ON order.book_id=book.book_id INNER JOIN  
user ON order.user_id=user.user_id;
```

→ Book,user ve order tablolarının joinlenmesinden elde edilen tablodan bookname ,date,amount ve subtotal değerleri çekilip ekrana bastırılır ve kullanıcının önceden vermiş olduğu siparişleri gösterilir.

## Future Extensions

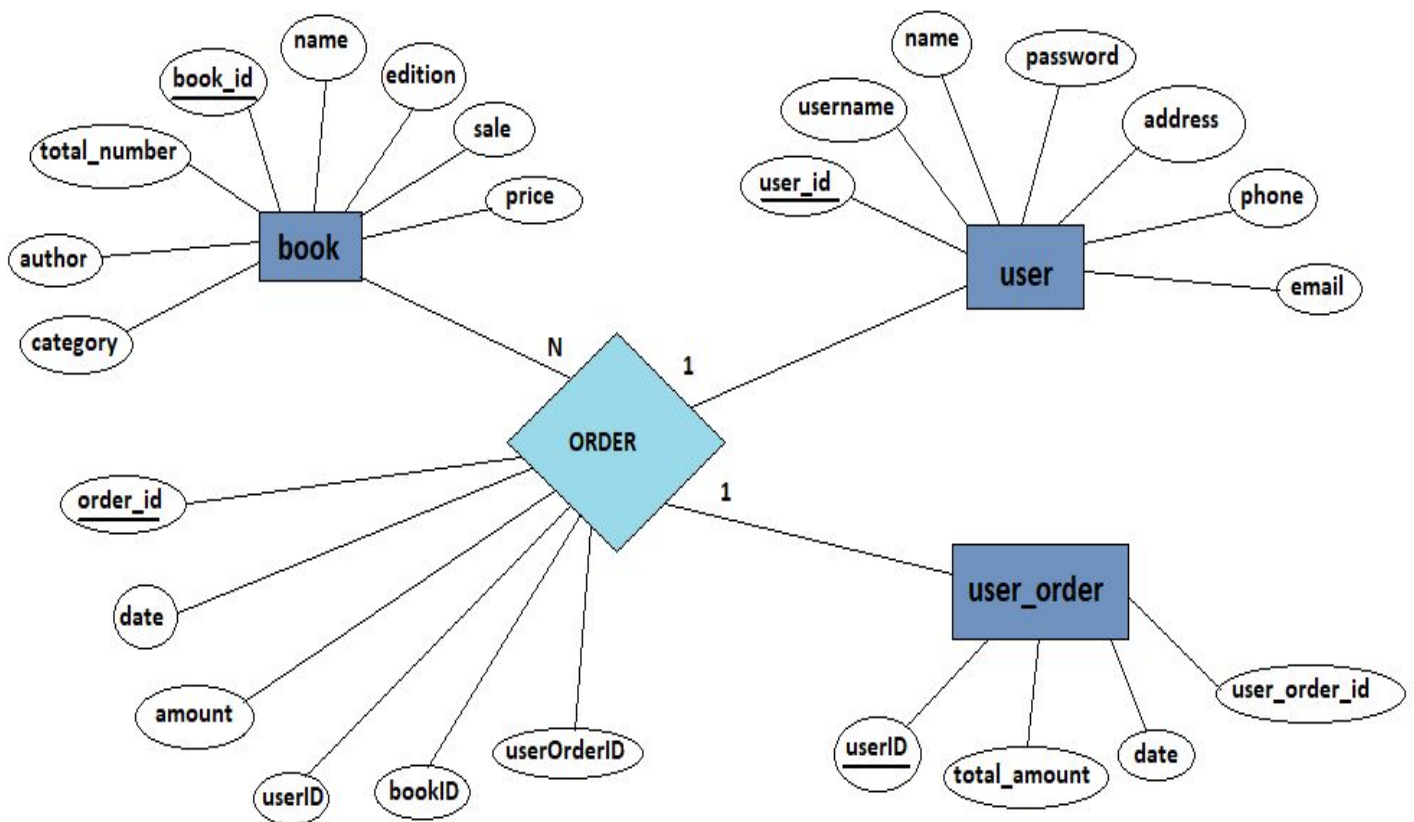
1. Geçmiş siparişler içinde arama, filtreleme yapmak.
2. Geçmişte alınan kitaplara göre kitap tavsiyesi oluşturmak.
3. Kategoriler için ayrı bir kategori tablosu tutulması.



# Appendix

## Appendix A - EER Diagram

BookStore EER Diagram



## Appendix B - DDL Script

```
CREATE TABLE book(  
    book_id INTEGER NOT NULL,  
    name VARCHAR(50) NOT NULL,  
    edition INTEGER NOT NULL,  
    sale INTEGER NOT NULL,  
    price FLOAT NOT NULL,  
    total_number INTEGER NOT NULL,  
    author VARCHAR(50) NOT NULL,  
    category VARCHAR(50) NOT NULL,  
    PRIMARY KEY (book_id)  
);  
CREATE TABLE "user"(  
    user_id INTEGER NOT NULL,  
    username VARCHAR(50) NOT NULL,  
    name VARCHAR(50) NOT NULL,  
    password VARCHAR(50) NOT NULL,  
    address VARCHAR(50) NOT NULL,  
    phone VARCHAR(50) NOT NULL,  
    email VARCHAR(50) NOT NULL,  
    PRIMARY KEY (user_id)  
);  
CREATE TABLE "order"(  
    order_id INTEGER NOT NULL,  
    "date" date,  
    amount FLOAT NOT NULL,  
    userID INTEGER NOT NULL,  
    bookID INTEGER NOT NULL,  
    userOrderID INTEGER NOT NULL,  
    book_name VARCHAR(100),  
    subtotal FLOAT,  
    PRIMARY KEY (order_id),  
    FOREIGN KEY(bookID) REFERENCES book(book_id),  
    FOREIGN KEY(userID) REFERENCES "user"(user_id)  
);  
CREATE TABLE user_order(  
    user_order_id INTEGER NOT NULL,  
    "date" date,  
    total_amount FLOAT NOT NULL,  
    userID INTEGER NOT NULL,  
    PRIMARY KEY (user_order_id),  
    FOREIGN KEY(userID) REFERENCES "user"(user_id)  
);
```

## Appendix C - Mini World

Sistemde kullanıcılar öncelikle üye olurlar sonra login olurlar.Kullanıcılar kitap order edebilirler,sepetlerine ekleyebilirler.Her kullanıcının bir ID , kullanıcı adı, isim, parola , adres, telefon numarası ve email bilgisi bulunur.Her kitabın ID, isim, baskı numarası, satış bilgisi, fiyat ,stok bilgisi, yazar ve kategori bilgisi bulunur.Kullanıcının verdiği her order için ID, tarih, miktar, kullanıcı sipariş ID, kitap adı ve sipariş ara toplam bilgisi bulunur.Kullanıcıların tek seferde verdiği siparişler için kullanıcı sipariş ID, tarih, toplam fiyat bilgisi tutulur.