Delivery book shop

Submitted to

Asst. Prof. Dr. Darun Kesrarat

Database management System
IT3210 Term Project

Submitted by

Palod Patimavirujh 5612602
Worachit Into 5614797
Kritanu Singhavorawuti 5616813
Woranat Swangswai 5837619
Objectives

Our DBMS is about delivery book rental system or we can call it as delivery library. We’re focusing on how to make customer feel convenient to easily get a book on their free time. So, we will create a website to allow customer register with our service. After Customers are registered with our service, they can choose any books that available on our website. If they are a real book worm, they will love in it! Renter can present a book to librarian for make an approval to show on the shop. Renter can communicate to the customer from the comment section. Moreover, we have provided an online librarian to support user Q&A in real time. But what’s an online librarian? Online librarian is a person who works as an admin of system which has authorize to manage our database system. Besides, they approve the book from Renter which must be appropriated to show on the website and follow the regulation of the website. There are few processes to use our service, customer just login, search for books, make payment, deliver and enjoy reading time!
Scope

There are 3 level of users involve in this system

1. Customer
2. Publisher
3. Librarian

All level of users is able to login to the system by using corresponding ID and password. Also provide specifically different ID Code for all users. So login/logout function is basic function for all users.

Customer

- Rent the book
  Customer can buy or rent the book directly on the website. Customer can buy in any time and also choose the period of time of rental of the book.

- View the book
  Customer can only view the short information about the book.

- Search the book from filter function
  Customer can find the book easily from filter system which provide the user a lot of kinds of book. Also the customer can know the book which related to the book that the user searches.

- Credit and debit card are provided
  Customer can pay by using Credit or Debit card.

- Contact to admin or Publisher
  Customer can contact to the admin if they are in trouble or they can contact to the Publisher if they have something about the book.

- View customer’s history
  Customer can check their rental history to check whether they’ve read the same book or not. So, they won’t waste their money and time.
Publisher

- *Add book*
  Publisher can add the book as they want.

- *Contact the admin’s and customer’s*
  Publisher can discussed with the admin and customer by using provided contract

Librarian (Admin)

- *Track the book*
  Admin can know which book belong to who and where is it now.

- *Summary of sales*
  Admin can know the summary of the sales in each month and each year.

- *View the user’s information*
  Librarian can see the information about the location of the customer and also the contract information.

- *View number of book that available*
  Publisher will check how many books that available on the stocks.

- *Contact the publisher’s and customer’s*
  Librarian can discuss with the publisher and customer by using provided contract.
CREATE TABLE ADMIN

(AdminID INTEGER GENERATED ALWAYS AS IDENTITY
  (START WITH 1
   INCREMENT BY 1
   MINVALUE 1
   MAXVALUE 1000000
   NO CYCLE),
  AdminName VARCHAR(30) NOT NULL,
  AdminPassword VARCHAR(10) NOT NULL,
  E-mail VARCHAR(30) NOT NULL,
  Contract NUMBER(10) NOT NULL,
CONSTRAINT Admin_PK PRIMARY KEY (AdminID));

CREATE TABLE Book

(BookID INTEGER GENERATED ALWAYS AS IDENTITY
  (START WITH 1
   INCREMENT BY 1
   MINVALUE 1
   MAXVALUE 1000000
   NO CYCLE),
  BookName VARCHAR(30) NOT NULL,
 TypeID NUMBER(10,0) NOT NULL,
  PublisherID NUMBER(10,0) NOT NULL,
  Quantity NUMBER(10,0) NOT NULL,
  Description VARCHAR(200),
CONSTRAINT Book_PK PRIMARY KEY (BookID);
CONSTRAINT Book_FK FOREIGN KEY (TypeID) REFERENCES BookType(TypeID);
CONSTRAINT Book_FK2 FOREIGN KEY (PublisherID) REFERENCES PublisherI(PublisherID));
CREATE TABLE BookType
    (TypeID NUMBER (10,0) NOT NULL,
     TypeName VARCHAR(30) NOT NULL,
    CONSTRAINT BookType_PK PRIMARY KEY (TypeID));

CREATE TABLE Customer
    (CusID INTEGER GENERATED ALWAYS AS IDENTITY
     (START WITH 1
      INCREMENT BY 1
      MINVALUE 1
      MAXVALUE 1000000
      NO CYCLE),
     CusName VARCHAR(30) NOT NULL,
     E-mail VARCHAR(30) NOT NULL,
     CusPassword NUMBER (10,0) NOT NULL,
     MobilePhone NUMBER(10,0) NOT NULL,
     CusAddress VARCHAR(40) NOT NULL,
    CONSTRAINT Customer_PK PRIMARY KEY (CusID));
CREATE TABLE DropPoint
(DropPointID INTEGER GENERATED ALWAYS AS IDENTITY
  (START WITH 1
   INCREMENT BY 1
   MINVALUE 1
   MAXVALUE 1000000
   NO CYCLE),
  Branch VARCHAR(30) NOT NULL,
  Location VARCHAR(30) NOT NULL,
  Contract NUMBER(10) NOT NULL,
CONSTRAINT DropPoint_PK PRIMARY KEY (DropPointID));

CREATE TABLE EMS_Shipment
(TrackingNumber INTEGER GENERATED ALWAYS AS IDENTITY
  (START WITH 101
   INCREMENT BY 1
   MINVALUE 1
   MAXVALUE 1000000
   NO CYCLE),
  OrderID NUMBER (10,0) NOT NULL,
CONSTRAINT EMS_Shipment_PK PRIMARY KEY (TrackingNumber));
CREATE TABLE OrderLine
(OrderID INTEGER GENERATED ALWAYS AS IDENTITY
(START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 1000000
NO CYCLE),
CusID NUMBER (10,0) NOT NULL,
OrderDate DATE DEFAULT SYSDATE,
PaymentID NUMBER (10,0) NOT NULL,
TotalPrice NUMBER (10,0) NOT NULL,
CONSTRAINT OrderLine_PK PRIMARY KEY (OrderID);
CONSTRAINT OrderLine_FK FOREIGN KEY (CusID) REFERENCES Customer(CusID);
CONSTRAINT OrderLine_FK2 FOREIGN KEY (PaymentID) REFERENCES Payment(PaymentID));
CREATE TABLE OrderDetail
(DetailID VARCHAR(20) NOT NULL,
BookID NUMBER (10,0) NOT NULL,
ReturnDate VARCHAR(20) NOT NULL,
DropPointID NUMBER (10,0) NOT NULL,
TrackingNumber NUMBER (10,0) NOT NULL,
Quantity NUMBER(10,0) NOT NULL,
Price NUMBER(10,0) NOT NULL,
OrderID NUMBER (10,0) NOT NULL,
AdminID NUMBER (10,0) NOT NULL,
Status VARCHAR(20) NOT NULL,
CONSTRAINT OrderLine_PK PRIMARY KEY (DetailID);
CONSTRAINT OrderLine_FK FOREIGN KEY (BookID) REFERENCES Book(BookID);
CONSTRAINT OrderLine_FK2 FOREIGN KEY (DropPointID) REFERENCES DropPoint (DropPointID);
CONSTRAINT OrderLine_FK3 FOREIGN KEY (TrackingNumber) REFERENCES EMS_Shipment(TrackingNumber);
CONSTRAINT OrderLine_FK4 FOREIGN KEY (OrderID) REFERENCES OrderLine(OrderID);
CONSTRAINT OrderLine_FK5 FOREIGN KEY (AdminID) REFERENCES Admin(AdminID));

CREATE TABLE Payment
(PaymentID NUMBER (10,0) NOT NULL,
Method VARCHAR(30) NOT NULL,
Credit VARCHAR(30) NOT NULL,
CONSTRAINT Payment_PK PRIMARY KEY (PaymentID));
CREATE TABLE Publisher
    (PublisherID INTEGER GENERATED ALWAYS AS IDENTITY
        (START WITH 1
         INCREMENT BY 1
         MINVALUE 1
         MAXVALUE 1000000
         NO CYCLE),
         Company VARCHAR(30) NOT NULL,
         Contract NUMBER (10,0) NOT NULL,
         Address VARCHAR(30) NOT NULL,
    CONSTRAINT Publisher_PK PRIMARY KEY (PublisherID));
SQL for retrieving data

Rental Book Store

Sign in
SELECT * FROM Customer WHERE Customer.E-mail = Username, Customer.CusPassword = Password;

Categories
SELECT * FROM BookType;

Book List
SELECT Book.BookName, BookType.TypeName, Book.Author, Book.Description, OrderDetail.Price, Book.Quantity
FROM (BookType INNER JOIN Book ON BookType.TypeID = Book.TypeID) INNER JOIN OrderDetail ON Book.BookID = OrderDetail.BookID;
**Search Book**
SELECT Book.BookName, BookType.TypeName, Book.Author, Book.Description, OrderDetail.Price, Book.Quantity
FROM (BookType INNER JOIN Book ON BookType.TypeID = Book.TypeID) INNER JOIN OrderDetail ON Book.BookID = OrderDetail.BookID
WHERE BookName = BookNameInput;

**Sign Up**
INSERT INTO Customer (CusName, E-mail, CusPassword, MobilePhone, CusAddress) VALUES (‘Name’, ‘E-mail’, ‘Password’, ‘MobileNumber’, ‘Address’);
Rental Book Store
Administration section

Admin Sign In
SELECT * FROM Admin WHERE AdminID = AdminID AND AdminPassword = Password;
Admin Info
SELECT AdminID, AdminName, E-mail, Contract WHERE AdminID = AdminID AND AdminPassword = AdminPassword

Drop point
SELECT DropPointId From DropPoint

Order Information
INNER JOIN Admin ON OrderDetail.AdminID = Admin.AdminID)
INNER JOIN OrderLine ON OrderDetail.OrderLineID = OrderLine.OrderLineID)
INNER JOIN Book ON OrderDetail.BookID = Book.BookID)
INNER JOIN DropPoint ON OrderDetail.DropPointID = DropPoint.DropPointID);
User Information
SELECT CusName, E-mail, MobilePhone FROM Customer

User Book Count
SELECT COUNT(OrderID) FROM OrderLine INNER JOIN Customer ON OrderLine.CustID = Customer.CustID
Publisher Information

SELECT Company, Address, Contract FROM Publisher WHERE PublisherID = PublisherID
**Add Book**

INSERT INTO Book (BookName, Author,TypeID,Description,Price) VALUES
(BookName,Author,TypeID,Description,Price)
User Address Information

SELECT CusAddress FROM Customer WHERE CusID = CusID
### Order Information

SELECT OrderDetail.OrderID, Book.BookName, OrderLine.OrderDate, OrderDetail.ReturnDate, OrderDetail.Quantity, OrderDetail.Price, OrderDetail.TrackingNumber
FROM ((OrderDetail
INNER JOIN Book ON OrderDetail.BookID = Book.BookID)
INNER JOIN OrderLine ON OrderDetail.OrderID = OrderLine.Order)
**Cart Information**

FROM ((OrderDetail INNER JOIN Book ON OrderDetail.BookID = Book.BookID)
INNER JOIN OrderLine ON OrderDetail.OrderID = OrderLine.OrderID)

**Payment Method**

SELECT Method FROM Payment
Book Information
FROM (Book
INNER JOIN BookType ON Book.TypeID = BookType.TypeID)
INNER JOIN Publisher ON Book.PublisherID = Publisher.PublisherID)
## Tracking Information

FROM ((OrderDetail INNERJOIN Book ON OrderDetail.BookID = Book.BookID) INNERJOIN DropPoint ON OrderDetail.DropPointID = DropPointID.DropPointID)

<table>
<thead>
<tr>
<th>Book ID</th>
<th>Book Name</th>
<th>Tracking No.</th>
<th>Book Transfer Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>#155</td>
<td>Book 1</td>
<td>E4473/24208TH</td>
<td>Received, Dispatch</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MUANG THAILAND COMPLEX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Bangkok EMS Centre</td>
</tr>
<tr>
<td>#155</td>
<td>Book 2</td>
<td>E4473/24209TH</td>
<td>Received, Dispatch</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MUANG THAILAND COMPLEX</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Bangkok EMS Centre</td>
</tr>
</tbody>
</table>
Publisher Information
SELECT Publisher.Company, Publisher.PublisherID, Publisher.Address, Publisher.Contract
WHERE Company = Publisher.Company