Term Project Deliverables

CSCE 4350

Faris hawamdeh

dustin fennessy

saeed babaker

ron galvan

2014

# Part 1: Data Attributes

## Books

* BookID
* Title
* Authors
* Genre
* Publisher
* Publication Date
* ISBN
* Format
* Price
* Comments
* Isdeleted

For deciding what information needed to be stored for books, we used the most likely information to be searched for from the Barnes and Noble website. We include the title, authors, genre, publisher, and ISBN as attributes. The publication date was added to provide a subcategory of books to allow for searching for newly released books or books soon to be released. To fulfil the requirements of part 5 we included user comments/reviews. The isdeleted attribute is added to simulate deletion from the web store but keep the data consistent in the database as to not create a situation where references to deleted books by other tables occur.

## Customer

* CustomerID
* Email
* Password
* Name
* Shipping Address
* Credit Card
* Isdeleted

For deciding what information to be stored for the customer, we decided on having the minimum information necessary for fulfilling an order. The email address and password were included as a way to contact the customer and identify them when they log in. The credit card is stored so that the customer can pay for the item. The name and shipping address is stored so that the item can shipped to the customer. The isdeleted attribute is added to simulate deletion from the web store but keep the data consistent in the database as to not create a situation where references to deleted customers by other tables occur.

## Seller

* SellerID
* Email
* Password
* Name
* Address
* Phone Number
* Items
* Isdeleted

For the seller information we adopted the strategy as the customer. Email address and password is stored to allow the seller to be contacted and identify them when they log in. Name, address, and phone number is stored in order for employees at the web store or customers can get in contact with seller if necessary. The database should also keep track of all the items being sold by that seller so that when orders are placed the orders can be directed to the seller so that they can fulfill them. The isdeleted attribute is added to simulate deletion from the web store but keep the data consistent in the database as to not create a situation where references to deleted sellers by other tables occur.

## Order

* OrderID
* CustomerID
* ShippingAddress
* Items
* Item Quantity
* Credit Card
* Cost
* Status

For orders we included the customerID so that the order is associated with a specific customer. The shipping address and credit card are also added in case the customer wishes to use a different credit card or ship the item to another address as a gift. The order should also contain the items that the customer wanted and the quantity of each item. The order also has a status to let the customer know if the order has been shipped yet.

## Shopping Cart

* CustomerID
* Cost
* Items
* Item Quantity

The shopping cart includes the customerID of the customer that the cart is associated with. The cart should contain the items that the customer wishes to purchase, the quantity of the items, and the cost of the total purchase.

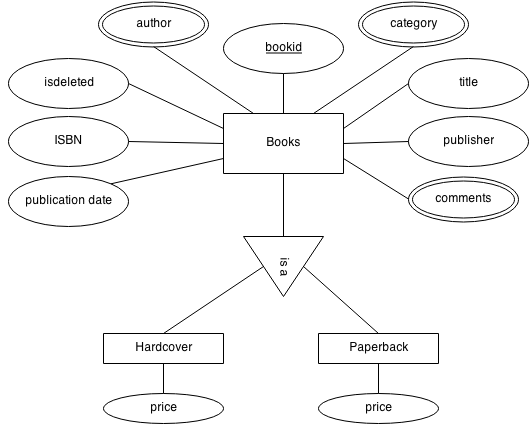
## Administrator

* AdminID
* Name
* Username
* Password
* isSuperAdmin

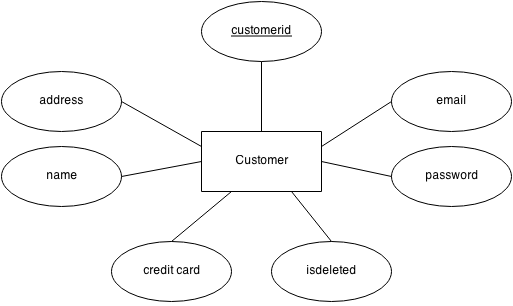
The administrator entity contains the name of the administrator. The username and password used to log in the administrator interface is also stored. The privilege level of the employee is also stored to identify whether they have the authority to create new administrator accounts.

# Part 2: ER Diagrams

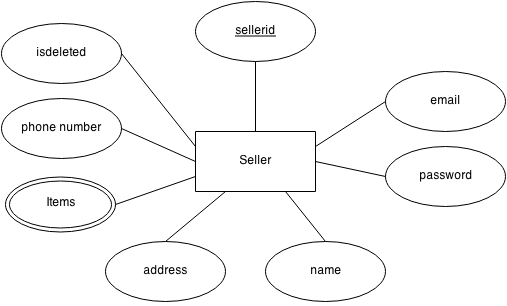
## Books Entity



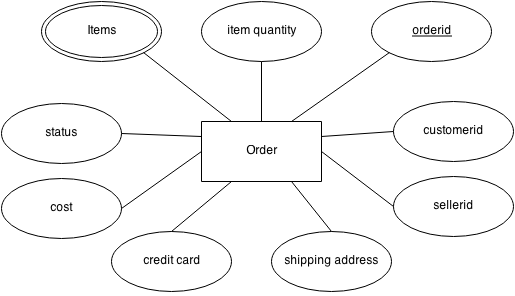
## Customer Entity



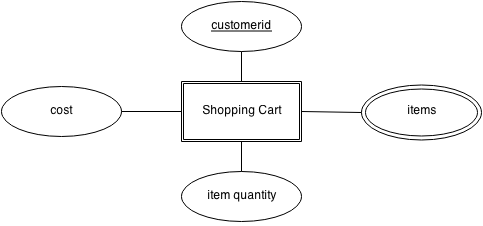
## Seller Entity



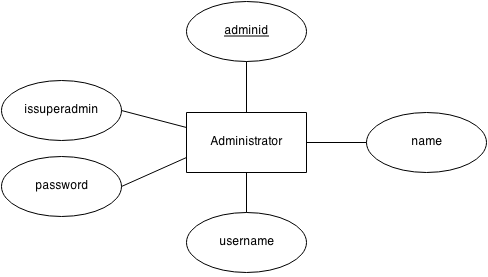
## Order Entity



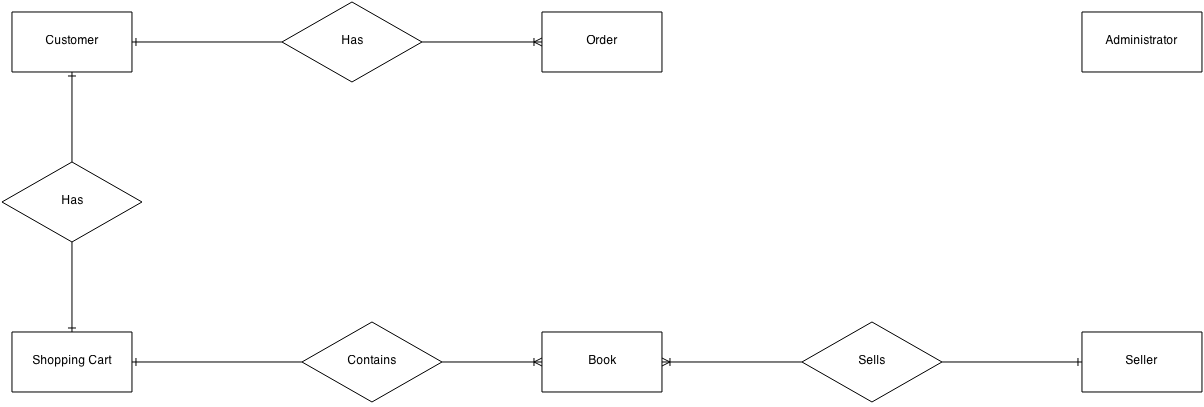
## Shopping Cart Entity



## Administrator Entity



## Full Relation



# Part 3: Tables

## Admin

CREATE TABLE IF NOT EXISTS `admin` (

`adminid` int(11) NOT NULL AUTO\_INCREMENT,

`firstname` varchar(45) NOT NULL,

`lastname` varchar(45) NOT NULL,

`username` varchar(45) NOT NULL,

`password` varchar(45) NOT NULL,

`issuper` tinyint(4) NOT NULL DEFAULT '0',

PRIMARY KEY (`adminid`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

## Author

CREATE TABLE IF NOT EXISTS `author` (

`bookid` int(11) NOT NULL,

`author` varchar(45) NOT NULL,

PRIMARY KEY (`bookid`,`author`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

## Book

CREATE TABLE IF NOT EXISTS `book` (

`bookid` int(11) NOT NULL AUTO\_INCREMENT,

`title` varchar(45) NOT NULL,

`publisher` varchar(45) NOT NULL,

`publicationdate` date DEFAULT NULL,

`isbn` varchar(17) DEFAULT NULL,

`isdeleted` tinyint(4) NOT NULL DEFAULT '0',

PRIMARY KEY (`bookid`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1 ;

## Book Format

CREATE TABLE IF NOT EXISTS `bookformat` (

`bookid` int(11) NOT NULL,

`format` varchar(20) NOT NULL,

`price` int(5) NOT NULL,

PRIMARY KEY (`bookid`,`format`,`price`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

## Category

CREATE TABLE IF NOT EXISTS `category` (

`bookid` int(11) NOT NULL,

`genre` int(10) NOT NULL,

PRIMARY KEY (`bookid`,`genre`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

## Comments

CREATE TABLE IF NOT EXISTS `comments` (

`bookid` int(11) NOT NULL,

`customerid` int(11) NOT NULL,

`date` datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`comment` text,

PRIMARY KEY (`bookid`,`customerid`,`date`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

## Customer

CREATE TABLE IF NOT EXISTS `customer` (

`customerid` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(45) NOT NULL,

`emailaddress` varchar(60) NOT NULL,

`password` varchar(45) NOT NULL,

`address` varchar(45) DEFAULT NULL,

`city` varchar(45) DEFAULT NULL,

`state` varchar(45) DEFAULT NULL,

`zip` varchar(7) DEFAULT NULL,

`creditcard` varchar(20) DEFAULT NULL,

`isdeleted` tinyint(4) NOT NULL DEFAULT '0',

PRIMARY KEY (`customerid`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1 ;

## Order Items

CREATE TABLE IF NOT EXISTS `orderitems` (

`orderid` int(11) NOT NULL,

`bookid` int(11) NOT NULL,

`format` varchar(20) NOT NULL,

`quantity` int(5) NOT NULL,

`status` tinyint(4) NOT NULL DEFAULT '0',

`cost` int(10) DEFAULT NULL,

PRIMARY KEY (`orderid`,`bookid`,`format`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

## Orders

CREATE TABLE IF NOT EXISTS `orders` (

`orderid` int(11) NOT NULL AUTO\_INCREMENT,

`customerid` int(11) NOT NULL,

`creditcard` varchar(20) DEFAULT NULL,

`address` varchar(45) DEFAULT NULL,

`city` varchar(45) DEFAULT NULL,

`state` varchar(45) DEFAULT NULL,

`zip` varchar(7) DEFAULT NULL,

`total` int(7) DEFAULT NULL,

PRIMARY KEY (`orderid`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=7;

## Seller

CREATE TABLE IF NOT EXISTS `seller` (

`sellerid` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(45) NOT NULL,

`emailaddress` varchar(60) NOT NULL,

`password` varchar(45) NOT NULL,

`phonenumber` varchar(12) NOT NULL,

`address` varchar(45) NOT NULL,

`city` varchar(45) NOT NULL,

`state` varchar(45) NOT NULL,

`zip` varchar(7) NOT NULL,

`isdeleted` tinyint(4) NOT NULL DEFAULT '0',

PRIMARY KEY (`sellerid`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1 ;

## Seller Item

CREATE TABLE IF NOT EXISTS `selleritem` (

`sellerid` int(11) NOT NULL,

`bookid` int(11) NOT NULL,

PRIMARY KEY (`sellerid`,`bookid`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

# Part 4: SQL Statements

## Insert Statements

INSERT INTO `admin` (`adminid`, `firstname`, `lastname`, `username`, `password`, `issuper`) VALUES

(1, 'John', 'Doe', 'Maou', 'password', 1),

(2, 'Jane', 'Doe', 'Haru', 'password', 0);

INSERT INTO `author` (`bookid`, `author`) VALUES

(1, 'James Patterson'),

(2, 'Jason Wilson'),

(2, 'Paul Theroux'),

(3, 'Gina Homolka'),

(4, 'Sean Brock'),

(5, 'The Editors of Southern Living Magazine'),

(6, 'Donna Tartt'),

(7, 'David Foster Wallace'),

(8, 'Joseph Heller'),

(9, 'James Patterson'),

(9, 'Marshall Karp');

INSERT INTO `book` (`bookid`, `title`, `publisher`, `publicationdate`, `isbn`, `isdeleted`) VALUES

(1, 'Hope to Die', 'Little, Brown and Company', '2014-11-24', '9780316210966', 0),

(2, 'The Best American Travel Writing 2014', 'Houghton Mifflin Harcourt', '2014-07-10', '9780544330153', 0),

(3, 'The Skinnytaste Cookbook', 'Crown Publishing Group', '2014-09-30', '9780385345620', 0),

(4, 'Heritage', 'Artisan', '2014-10-21', '9781579654634', 0),

(5, 'The Southern Cake Book', 'Oxmoor House', '2014-05-27', '9780848702984', 0),

(6, 'The Goldfinch', 'Little, Brown and Company', '2013-10-22', '9780316055437', 0),

(7, 'Infinite Jest', 'Little, Brown and Company', '2006-11-13', '9780316066525', 0),

(8, 'Catch-22', 'Simon & Schuster', '1951-04-05', '9781451626650', 0),

(9, 'NYPD Red 3', 'Little, Brown and Company', '2015-03-15', '9780316406994', 0);

INSERT INTO `bookformat` (`bookid`, `format`, `price`) VALUES

(1, 'Hardcover', 15),

(1, 'Paperback', 10),

(2, 'Hardcover', 15),

(2, 'Paperback', 12),

(3, 'Hardcover', 18),

(4, 'Hardcover', 25),

(5, 'Hardcover', 30),

(5, 'Paperback', 20),

(6, 'Hardcover', 19),

(7, 'Hardcover', 22),

(7, 'Paperback', 14),

(8, 'Hardcover', 20),

(8, 'Paperback', 10),

(9, 'Hardcover', 20),

(9, 'Paperback', 16);

INSERT INTO `category` (`bookid`, `genre`) VALUES

(1, 1),

(1, 3),

(2, 0),

(3, 4),

(4, 4),

(5, 4),

(6, 3),

(7, 2),

(7, 3),

(8, 2),

(8, 3),

(9, 1),

(9, 4);

INSERT INTO `comments` (`bookid`, `customerid`, `date`, `comment`) VALUES

(1, 1, '2014-11-28 08:27:06', 'This book is Boring!!!\r\nLOOOOOL!!!');

INSERT INTO `customer` (`customerid`, `name`, `emailaddress`, `password`, `address`, `city`, `state`, `zip`, `creditcard`, `isdeleted`) VALUES

(1, 'Faris Hawamdeh', 'farishawamdeh@gmail.com', 'password', '1155 Union Circle', 'Denton', 'Texas', '76203', '1111222233334444', 0),

(2, 'Jason Lindsey', 'JasonSLindsey@armyspy.com', 'ich1ul0ohhiK', '1180 Oak Way', 'Lincoln', 'Nebraska', '68501', '4716992911218985', 0),

(3, 'Martin Winter', 'MartinDWinter@dayrep.com', 'EiW8voeP6', '2412 Marion Drive', 'Tampa', 'Florida', '33634', '5113979318306980', 0),

(4, 'Elizabeth Jones', 'ElizabethJJones@armyspy.com', 'feCh0quee0Ee', '6 Barrington Court', 'Rector', 'Arkansas', '72461', '5139696654966991', 0),

(5, 'Charles Dooley', 'CharlesVDooley@jourrapide.com', 'Gie9ohPai', '3594 Elk Rd Little', 'Tucson', 'Arizona', '85716', '4916697506926805', 0);

INSERT INTO `orderitems` (`orderid`, `bookid`, `format`, `quantity`, `status`, `cost`) VALUES

(1, 1, 'Hardcover', 1, 0, 15),

(2, 1, 'Paperback', 2, 0, 20),

(2, 3, 'Hardcover', 1, 0, 18),

(3, 8, 'Paperback', 1, 0, 10),

(4, 4, 'Hardcover', 1, 0, 25),

(4, 5, 'Hardcover', 1, 0, 30),

(5, 5, 'Paperback', 1, 0, 20),

(6, 5, 'Hardcover', 1, 0, 30);

INSERT INTO `orders` (`orderid`, `customerid`, `creditcard`, `address`, `city`, `state`, `zip`, `total`) VALUES

(1, 1, '1111222233334444', '1155 Union Circle', 'Denton', 'Texas', '76203', 15),

(2, 2, '4716992911218985', '1180 Oak Way', 'Lincoln', 'Nebraska', '68501', 58),

(3, 4, '5139696654966991', '6 Barrington Cour', 'Rector', 'Arkansas', '72461', 10),

(4, 5, '4916697506926805', '3594 Elk Rd Little', 'Tucson', 'Arizona', '85716', 55),

(5, 3, '5113979318306980', '2412 Marion Drive', 'Tampa', 'Florida', '33634', 20),

(6, 1, '1111222233334444', '1155 Union Circle', 'Denton', 'Texas', '76203', 30);

INSERT INTO `seller` (`sellerid`, `name`, `emailaddress`, `password`, `phonenumber`, `address`, `city`, `state`, `zip`, `isdeleted`) VALUES

(1, 'Bookz!', 'admin@bookz.com', 'password', '9402480021', '2520 Rodeo Plaza', 'Fort Worth', 'Texas', '76164', 0),

(2, 'Elizabeth Jones', 'ElizabethJJones@armyspy.com', 'feCh0quee0Ee', '9708540373', '1170 Stark Hollow Road', 'Holyoke', 'Colorado', '80734', 0);

INSERT INTO `selleritem` (`sellerid`, `bookid`) VALUES

(1, 1),

(1, 2),

(1, 3),

(1, 7),

(1, 8),

(1, 9),

(2, 4),

(2, 5),

(2, 6);

## Misc. Statements

1. List all the products posted by a particular seller, Elizabeth Jones.

SELECT title, publisher, publicationdate, isbn FROM book

INNER JOIN

(SELECT bookid FROM selleritem where sellerid=(SELECT sellerid FROM `seller` WHERE name='Elizabeth Jones')) as t ON book.bookid=t.bookid

| **title** | **publisher** | **publicationdate** | **isbn** |
| --- | --- | --- | --- |
| Heritage | Artisan | 2014-10-21 | 9781579654634 |
| The Southern Cake Book | Oxmoor House | 2014-05-27 | 9780848702984 |
| The Goldfinch | Little, Brown and Company | 2013-10-22 | 9780316055437 |

1. List all the products in the Mystery Genre( genre = 1 ) bought by customer Faris Hawamdeh.

SELECT title, publisher, publicationdate, isbn FROM book

INNER JOIN

(SELECT a.bookid FROM

(SELECT d.bookid, genre from category

INNER JOIN

(SELECT bookid FROM orderitems

INNER JOIN

(SELECT orderid FROM orders WHERE customerid=(SELECT customerid FROM customer WHERE name='Faris Hawamdeh')) as t ON orderitems.orderid=t.orderid) as d ON category.bookid=d.bookid) as a

WHERE genre=1) as k ON book.bookid=k.bookid

| **title** | **publisher** | **publicationdate** | **isbn** |
| --- | --- | --- | --- |
| Hope to Die | Little, Brown and Company | 2014-11-24 | 9780316210966 |

1. List the most popular (most copies sold) of one of your products.

SELECT a.title, a.publisher, a.publicationdate, a.isbn, SUM(a.quantity) FROM

(SELECT book.bookid, book.title, book.publisher, book.publicationdate, book.isbn, orderitems.quantity FROM book

INNER JOIN orderitems ON book.bookid=orderitems.bookid) as a

GROUP BY a.bookid ORDER BY SUM(a.quantity) DESC

| **title** | **publisher** | **publicationdate** | **isbn** | **SUM(a.quantity)** |
| --- | --- | --- | --- | --- |
| The Southern Cake Book | Oxmoor House | 2014-05-27 | 9780848702984 | 3 |
| Hope to Die | Little, Brown and Company | 2014-11-24 | 9780316210966 | 3 |
| Catch-22 | Simon & Schuster | 1951-04-05 | 9781451626650 | 1 |
| The Skinnytaste Cookbook | Crown Publishing Group | 2014-09-30 | 9780385345620 | 1 |
| Heritage | Artisan | 2014-10-21 | 9781579654634 | 1 |

1. Find the total amount of profit made from a customer, Elizabeth Jones in the one of your categories. Profit, in this context means the difference between how much money Carol spent and how much money she made. So, if Carol sold 5 records at $15 each and bought 2 records at $5 each, she made a profit of $65.

SELECT SUM(x.cost) - SUM(y.total) FROM

(SELECT cost FROM orderitems

INNER JOIN

(SELECT \* FROM selleritem WHERE sellerid=

(SELECT sellerid FROM seller WHERE name='Elizabeth Jones')) as a ON orderitems.bookid=a.bookid) as x,

(SELECT total FROM orders WHERE customerid=

(SELECT customerid FROM customer WHERE name='Elizabeth Jones')) as y

| **SUM(x.cost) - SUM(y.total)** |
| --- |
| 65 |