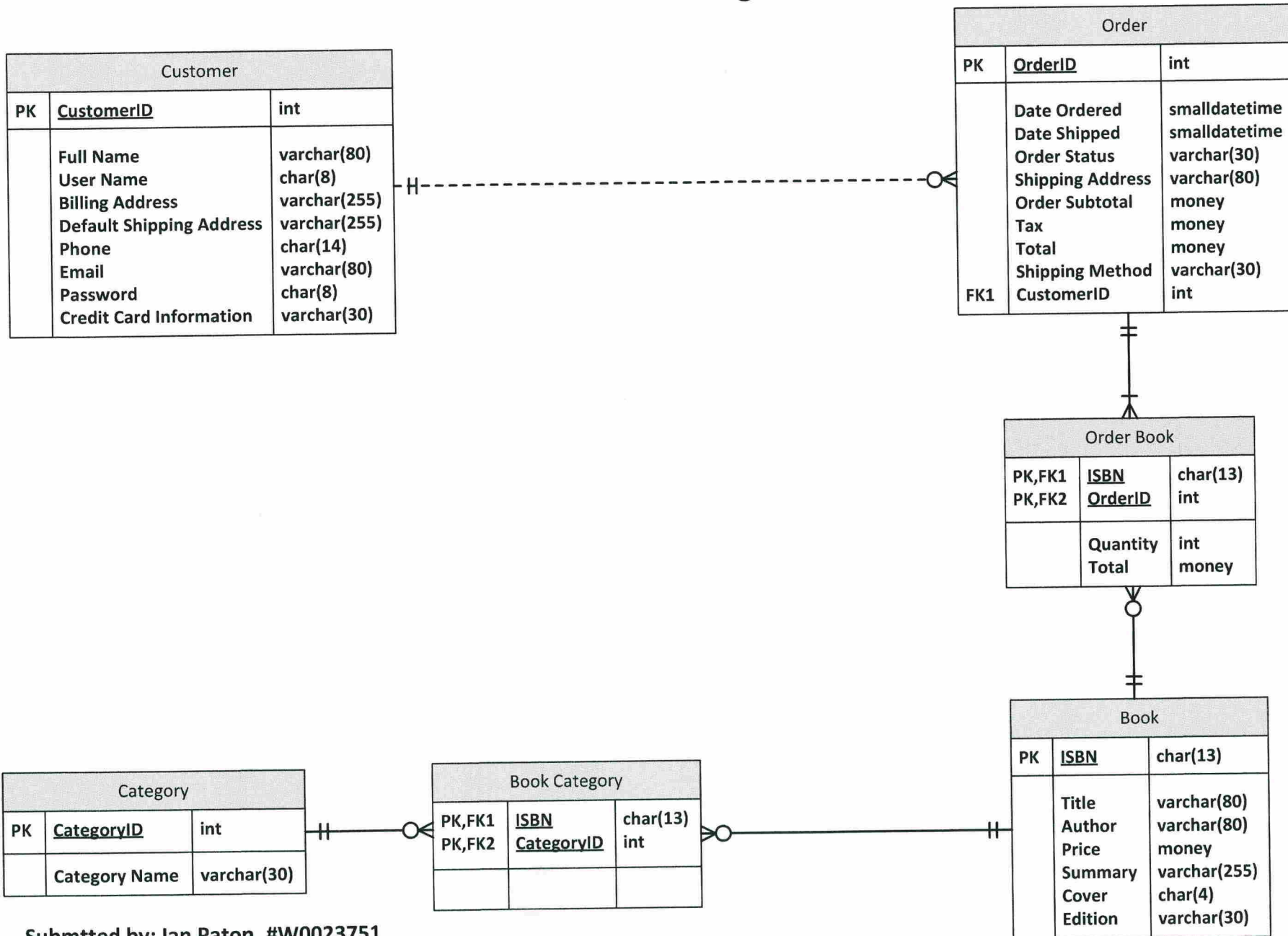


Online Book Retailer ~ Jungle.com

ERD Diagram



Submitted by: Ian Paton, #W0023751

Submitted by: Michael Petry, #0202030

```

--create a new database

CREATE DATABASE junglecom;
GO

--CREATE DATABASE junglecom

--use the new database
USE junglecom;
GO

--Create table customers
CREATE TABLE Customer
(
CustomerID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
[FullName] VARCHAR(80) NOT NULL,
UserName CHAR(8) NOT NULL,
BillingAddress VARCHAR(255) NOT NULL,
DefaultBillingAddress VARCHAR(255) NOT NULL,
Phone CHAR(14) NOT NULL,
Email VARCHAR(80) NOT NULL,
Password CHAR(8) NOT NULL,
CreditCardInformation VARCHAR(30) NOT NULL
);
GO

--create the Order table
CREATE TABLE [Order]
(
OrderID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
DateOrdered SMALLDATETIME NOT NULL,
DateShipped SMALLDATETIME NOT NULL,
OrderStatus VARCHAR(30) NOT NULL,
ShippingAddress VARCHAR(80) NOT NULL,
OrderSubtotal MONEY NOT NULL,
Tax MONEY NOT NULL,
OrderTotal MONEY NOT NULL,
ShippingMethod VARCHAR(30) NOT NULL,
CustomerID INT NOT NULL
);
GO

--create the books table

CREATE TABLE Book
(
ISBN CHAR(13) NOT NULL PRIMARY KEY,
Title VARCHAR(80) NOT NULL,
Author VARCHAR(80) NOT NULL,
Price MONEY NOT NULL,
Summary VARCHAR(255) NOT NULL,
Cover CHAR(4) NOT NULL,
Edition VARCHAR(30) NOT NULL
);
GO

--create Categories Table
CREATE TABLE Category
(
CategoryID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
CategoryName VARCHAR(30) NOT NULL
);
GO

--create Order Item Table
CREATE TABLE OrderBook

```

```

{
OrderID INT NOT NULL,
ISBN CHAR(13) NOT NULL,
Quantity INT NOT NULL,
Total MONEY NOT NULL
CONSTRAINT PK_OrderBook
PRIMARY KEY (OrderID, ISBN)
};
GO

--create book category table
CREATE TABLE BookCategory
(
ISBN CHAR(13) NOT NULL,
CategoryID INT NOT NULL
CONSTRAINT PK_BookCategory
PRIMARY KEY (ISBN, CategoryID)
);
GO

--add constraint between BookCategory and parent tables

ALTER TABLE BookCategory
ADD
CONSTRAINT FK_BookCategory_Category
FOREIGN KEY (CategoryID)
REFERENCES Category(CategoryID) ON DELETE CASCADE,
CONSTRAINT FK_BookCategory_Book
FOREIGN KEY (ISBN)
REFERENCES Book(ISBN) ON DELETE CASCADE;
GO

--add constraint between OrderBooks and parent tables

ALTER TABLE OrderBook
ADD
CONSTRAINT FK_OrderBook_Book
FOREIGN KEY (ISBN)
REFERENCES Book(ISBN) ON DELETE CASCADE,
CONSTRAINT FK_OrderBook_Order
FOREIGN KEY (OrderID)
REFERENCES [Order](OrderID) ON DELETE CASCADE;
GO

-- add a constraint between order and customer tables

ALTER TABLE [Order]
ADD
CONSTRAINT FK_Order_Customer
FOREIGN KEY (CustomerID)
REFERENCES Customer(CustomerID) ON DELETE CASCADE;
GO

-- add three books and their information to the database

INSERT INTO Book (ISBN, Title, Author, Price, Summary, Cover, Edition)
VALUES ('1234567890123', 'Beginning Programming for Dummies', 'Wallace Wang', '20.99',
'This books shows you how computer programming woks without all the technical details
or hard programming language.', 'Soft', 'Third Edition')
GO

INSERT INTO Book (ISBN, Title, Author, Price, Summary, Cover, Edition)
VALUES ('1234567890124', 'Web Design: Introductory Concepts and Techniques', 'Steven

```

```

        Shelly', '64.90', 'This book explains the connection between a detailed design plan
        that considers audience needs, site purpose, and various technical issues of a
        successful website.', 'Soft', 'Fifth Edition')
GO

INSERT INTO Book (ISBN, Title, Author, Price, Summary, Cover, Edition)
VALUES ('1234567890125', 'Network Security: Private Communication in a public world',
        'Mark Kaufmann', '56.23', 'This book brings together clear, insightful, and clever
        explanations of every key facet of information security.', 'Soft', 'Fourth Edition')

-- create customer information for three fictitious people

INSERT INTO Customer ([FullName], UserName, BillingAddress, DefaultBillingAddress, Phone
, Email, Password, CreditCardInformation)
VALUES ('Paul Haans', 'phaans70', '485 Champagne Drive, North York, ON M3J 2C6', '485
        Champagne Drive, North York, ON M3J 2C6', '(416) 635-2936', 'phaans23@gmail.com',
        'abcd1234', '8888-9999-9200-2192 12/12')
GO

INSERT INTO Customer ([FullName], UserName, BillingAddress, DefaultBillingAddress, Phone
, Email, Password, CreditCardInformation)
VALUES ('Jeremy Johnson', 'jeremy99', '46 Rivalda Road, North York, ON M9M 2M3', '46
        Rivalda Road, North York, ON M9M 2M3', '(416) 745-8078', 'jjohnson78@yahoo.ca',
        'y2h8k8s6', '7777-2222-4500-2398 10/12')
GO

INSERT INTO Customer ([FullName], UserName, BillingAddress, DefaultBillingAddress, Phone
, Email, Password, CreditCardInformation)
VALUES ('Samantha Murphy', 'sammy123', '49 Laird Drive, Toronto, ON M4G 3T4', '49 Laird
        Drive, Toronto, ON M4G 3T4', '(416) 429-2569', 'sammy_murphy2@hotmail.com', '1234lolk
        ', '5555-6666-8700-2891 09/14')
GO

-- create three new orders

-- one customer buys one copy of all three books
INSERT INTO [Order] (DateOrdered, DateShipped, OrderStatus, ShippingAddress,
        OrderSubtotal, Tax, OrderTotal, ShippingMethod, CustomerID)
VALUES ('2010-03-12', '2010-03-14', 'Not Yet Shipped', '485 Champagne Drive, North York,
        ON M3J 2C6', '142.12', '18.47', '160.59', 'Fast & Free Delivery', '1')
GO

INSERT INTO [OrderBook] (OrderID, ISBN, Quantity, Total)
VALUES ('1', '1234567890123', '1', '20.99')
GO

INSERT INTO [OrderBook] (OrderID, ISBN, Quantity, Total)
VALUES ('1', '1234567890124', '1', '64.90')
GO

INSERT INTO [OrderBook] (OrderID, ISBN, Quantity, Total)
VALUES ('1', '1234567890125', '1', '56.23')
GO

-- one customer buys just one of the books

INSERT INTO [Order] (DateOrdered, DateShipped, OrderStatus, ShippingAddress,
        OrderSubtotal, Tax, OrderTotal, ShippingMethod, CustomerID)
VALUES ('2010-03-16', '2010-03-19', 'Completed', '49 Laird Drive, Toronto, ON M4G 3T4',
        '20.99', '3.14', '24.13', 'Fast & Free Delivery', '3')
GO

INSERT INTO [OrderBook] (OrderID, ISBN, Quantity, Total)
VALUES ('2', '1234567890123', '1', '20.99')
GO

```

```

-- one customer buys five copies of one book, two copies of the second and ten copies of
the third

INSERT INTO [Order] (DateOrdered, DateShipped, OrderStatus, ShippingAddress,
OrderSubtotal, Tax, OrderTotal, ShippingMethod, CustomerID)
VALUES ('2010-03-17', '2010-03-22', 'Not Yet Shipped', '91 Dundas Square, Toronto, ON M4Y
1X7', '797.05', '119.55', '916.60', 'Fast & Free Delivery', '2')
GO

INSERT INTO [OrderBook] (OrderID, ISBN, Quantity, Total)
VALUES ('3', '1234567890123', '5', '104.95')
GO

INSERT INTO [OrderBook] (OrderID, ISBN, Quantity, Total)
VALUES ('3', '1234567890124', '2', '129.80')
GO

INSERT INTO [OrderBook] (OrderID, ISBN, Quantity, Total)
VALUES ('3', '1234567890125', '10', '562.30')
GO

-- insert sample data into category

INSERT INTO [Category] (CategoryName)
VALUES ('Programming');
GO

INSERT INTO [Category] (CategoryName)
VALUES ('Networking');
GO

INSERT INTO [Category] (CategoryName)
VALUES ('Web Development');
GO

INSERT INTO [Category] (CategoryName)
VALUES ('Web Design');
GO

INSERT INTO [Category] (CategoryName)
VALUES ('Database');
GO

-- assign each of the three books in the database a category

INSERT INTO BookCategory (ISBN, CategoryID)
VALUES ('1234567890123', '1');
GO

INSERT INTO BookCategory (ISBN, CategoryID)
VALUES ('1234567890125', '2');
GO

INSERT INTO BookCategory (ISBN, CategoryID)
VALUES ('1234567890125', '5');
GO

INSERT INTO BookCategory (ISBN, CategoryID)
VALUES ('1234567890124', '3');
GO

INSERT INTO BookCategory (ISBN, CategoryID)
VALUES ('1234567890124', '4');
GO

-- delete the second order

```

```

DELETE FROM [Order]
WHERE OrderID = 2;
GO

-- change the status of the first order to complete

UPDATE [Order]
SET OrderStatus = 'Completed'
WHERE OrderID = '1';
GO

-- change the third order to add another copy of the second book

UPDATE [OrderBook]
SET Quantity = '3'
WHERE OrderID = '3' AND ISBN = '1234567890124';

--Display all customer records

SELECT * FROM Customer
GO

--Display the title, Author, ISBN, price related to all
--books that have Database as a category or in the name of
--book.

SELECT * FROM [Order]
INNER JOIN OrderBook
ON [Order].OrderID = OrderBook.OrderID
INNER JOIN Book
ON Book.ISBN = OrderBook.ISBN
WHERE Quantity > '1';

--Display the title, Author, ISBN, price related to all
--books that have Database as a category or in the name of
--book.

SELECT Book.[Title], Book.[Author], Book.[ISBN], Book.[Price] FROM Book
INNER JOIN BookCategory
ON Book.ISBN = BookCategory.ISBN
INNER JOIN Category
ON Category.CategoryID = BookCategory.CategoryID
WHERE Title LIKE '%Database%' OR Summary LIKE '%Database%' OR CategoryName LIKE '%Database%'
GO

-- Display the email address of customer who have outstanding orders

SELECT Customer.Email FROM [Customer]
LEFT JOIN [Order]
ON Customer.CustomerID = [Order].CustomerID
WHERE [Order].OrderStatus <> 'Completed';

-- display all orders that have purchased more than one copy of any of the books

SELECT * FROM [Order]
INNER JOIN OrderBook
ON [Order].OrderID = OrderBook.OrderID
INNER JOIN Book
ON Book.ISBN = OrderBook.ISBN
WHERE Quantity > '1';

-- display the number of the order and the total number of books in each order.

```

```

SELECT [Order].OrderID, SUM(Quantity) as [Tot. No. of Books] from [Order]
INNER JOIN OrderBook
ON [Order].OrderID = OrderBook.OrderID
INNER JOIN Book
ON Book.ISBN = OrderBook.ISBN
GROUP BY [Order].OrderID;
GO

-- display the total cost of the order by adding the cost of all items

SELECT [Order].OrderID, SUM(Total) as [Tot. No. of Books] from [Order]
INNER JOIN OrderBook
ON [Order].OrderID = OrderBook.OrderID
INNER JOIN Book
ON Book.ISBN = OrderBook.ISBN
GROUP BY [Order].OrderID;
GO

--delete the foreign key constraints

ALTER TABLE [Order]

DROP

    CONSTRAINT FK_Order_Customer;
GO

ALTER TABLE BookCategory

DROP

    CONSTRAINT FK_BookCategory_Category,
    CONSTRAINT FK_BookCategory_Book;

GO

ALTER TABLE OrderBook

DROP

    CONSTRAINT FK_OrderBook_Book,
    CONSTRAINT FK_OrderBook_Order;

GO

--Delete all junglecom tables

DROP TABLE Customer;
GO
DROP TABLE [Order];
GO
DROP TABLE Book;
GO
DROP TABLE Category;
GO
DROP TABLE BookCategory;
GO
DROP TABLE OrderBook;

GO

--delete jungle database

USE master;
GO

```

```
DROP DATABASE Junglecom;  
GO
```