OPIM 5272: BUSINESS PROCESS MODELING AND DATA MANAGEMENT

Business Process of Barnes & Noble's Sales Process Model

Team 5

Submitted by:

Yanfu Zhu

Mayank Makan

Kunja Dutta

Hui Huang

Ankit Dsouza

Content

1 Business Process of Barnes & Noble's Sales Process Model

1.1 Overview

Traditional book selling through stores witnessed a change in 1997 when electronic book retailing hit the market. In the United States, retail bookstore market leader Barnes & Noble faced competition from Amazon.com in the cyberspace.

US consumers spent $26 billion on books by 1996. This meant an average of 10 books were purchased by each U.S. citizen. Each year around 50,000 new titles were published.

Following the entry of amazon.com is the book retail industry, Barnes & Noble introduced barnesnadnoble.com. Earlier Border was its biggest competitor but Barnes & Noble had to change its operations and marketing strategies to survive the current competition. Competition between amazon and Barnes & Noble became intense and titles were rapidly added to each site. In order to lead, each company targeted particular categories.

1.2 Current Process Model

For the old process, which starts with the goal to increase Barnes & Noble's market share in book industry, there are two main things needed to do. One is the store managers order the bestselling books from publishers to meet customers' demands. Another is the marketing managers work out the marketing strategies to improve the sales. Although their approaches and steps are different, both things are served to satisfy the goal to improve market share. So there is no doubt that this is a business process. Below is description of this business process. What's more, the Swim Lane Diagram of the As – Is Sales Process has been modeled in Figure 1.

- The goal to increase market share triggers the starts of bestselling books' order and marketing steps.
- Store managers order the books that have good sales history from publishers. And publisher would choose to whether ship the books from itself or distributors, depend on the distances from the warehouse of Barnes & Noble to publisher or distributors. If there are distributors near BN's warehouse, the distributors would bulk, pack, and send the books to warehouse. If not, publisher itself would do these. After warehouse receive and send the ordered books, the WINGS, which is focused on inventory tracking, would update the inventory information. And stores would receive those books.

- Marketing managers collect the sales data, analyze customer behavior, and thus make sales strategies offered to the stores.
- After both receiving the books ordered and marketing strategies needed to implement, the stores would sell books to customers with customize marketing strategies.

From the above description, we can find there are some imperfect points making this process redundant and inefficient. Those are what we should clearly define and improve later.

1.3 Problems with current process

Although Barnes & Nobles had kept strong in its marketplace for past few years, which prove its success of management method, it didn't get aware of the potential problem associate with the changing environment. However, with the rise of Amazon and the revolution it brought, Barnes & Noble's shortcoming became more and more obvious. Following are main problems of current process for Barnes & Noble.

- **Lagging behind current market trends**
  Amazon had analyzed over 20 products on the basis of a number of criteria-including the size of the market, the number of SKUs, and traditional margins, distribution patterns, and competitors as well as how value could be added relative to the traditional business models by the internet. But Barnes & Nobles was still trapped in their habit of expansion based on their old strategy rather than scientifically analyze their product and market, and know what their customers potentially demand.
- **Inefficient traditional supply chain management**
  By developing its own website and information system, Amazon had successfully increase the efficiency in procurement and store operation. However, Barnes & Nobles had to spend much time and money dealing with publishers and distributors, which add large cost on customers' purchase.
- **Complex process for customers**
  Making good use of the internet, it is very convenient for customers to find books, book information, reviews and recommendations, and to place orders with private information well protected. On the opposite, Barnes & Nobles didn't take steps further and innovatively in those function, which made their customers cost much time and energy buying their target books.

- **Lack of effective information management**

  First, the WINGS, which keep track of the inventory, had to update the inventory data about stock, input, and output, every time there are books coming or shipping. And it also needed to report these inventory data to management. Second, the stores also need to collect its sales data and customer information, and report these data to marketing management. Third, marketing managers were faced with more tough problems because they should collect and classify sales data, analyze customers' purchase behavior, and make accurate marketing strategies.

- **Hard to receive Customers' feedback**

  Feedback is too important to be neglected because bestselling books do not work and are not so customized sometimes. But customers just purchase book and then go home. So it is not easy for the stores to know the feedback about these books. The result is that stores cannot get enough information about these books' popularity. And thus they may input the books disliked by customers.

1.4 Improving Current Process

- **Take Business Online**

  According to us, Barnes and Nobles should expand his business by starting a website and take their business online. From this Customer can take advantage of wide range titles B&N has to offer. To enable this, B&N have to upgrade their Information Technology

- **Provide better supply chain system**

  With online order system, B&N can efficiently track customer demands. B&N can order book from distributors keeping in mind demands of customer, helping them save Inventory space.

- **Easing the shopping process**

  B&N should provide home delivery of books. It will be easy for customer to order books online and getting delivered at home and it will save their time. It will surely help B&N to increase their customer base.

- **Improve reviewing system**

  B&N should include reviewing page on their website. With this customer will be able to give real time review of books and will help other customers to make selection. It's a high

possibility that customer will buy book from their website after reading the reviews, which will boost B&N sales.

As a result, the Swim Lane Diagram of the To – Be Sales Process has been modeled in Figure 1.

1.5 Business Needs

Customer data base, inventory data base, books data base, sales data base need to be maintained. WINGS is used for online inventory tracking. Bookmaster is used for improved store system. It fastens transactions, communication between different stores, distribution centers, wholesalers. More than 2.5 million book titles are maintained using Bookmaster which helps to keep track of books.

In superstores customers had to be maintained and new customers had to be attracted. This can be done by implementing a strong operations plan with the help of the publishers. Marketing strategies are also strengthened. Bestsellers and new edition books are kept in the front line. Extra services are offered in the stores such as serving coffee, after signing up with Starbucks, pulling in authors for talks and book signing.

Operations for the online store has a new warehouse implemented. Marketing strategy for online sales are of utmost importance. Firefly, AOL and referral programs helped to strengthen the marketing strategies. Firefly helps with online recommendation of books for customers who review the books. AOL provided online services to promote Barnes and Noble's online sales. Online bookstores also offered customer reviews.

Barnes and Noble aims to lead the competition by efficiently maintaining the required databases and steadily following the processes.

Figure 1:

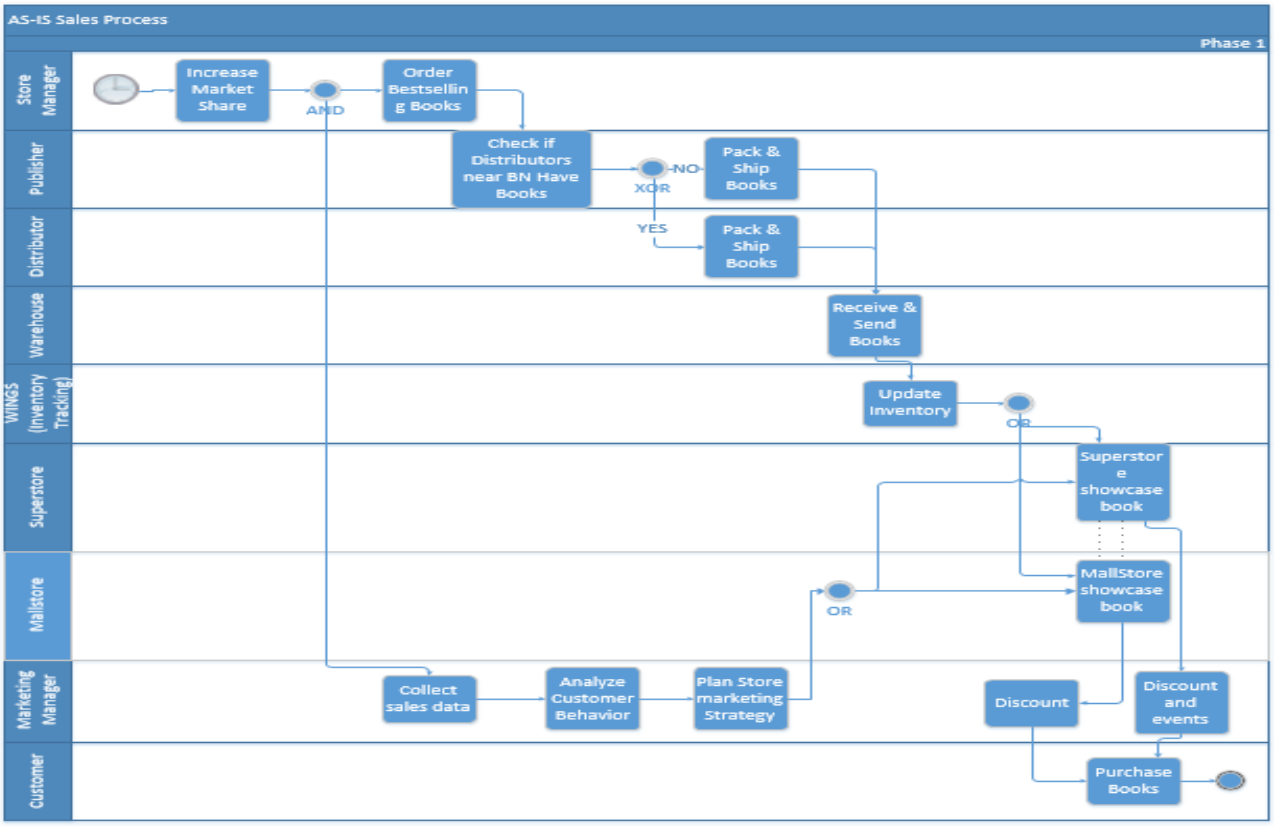
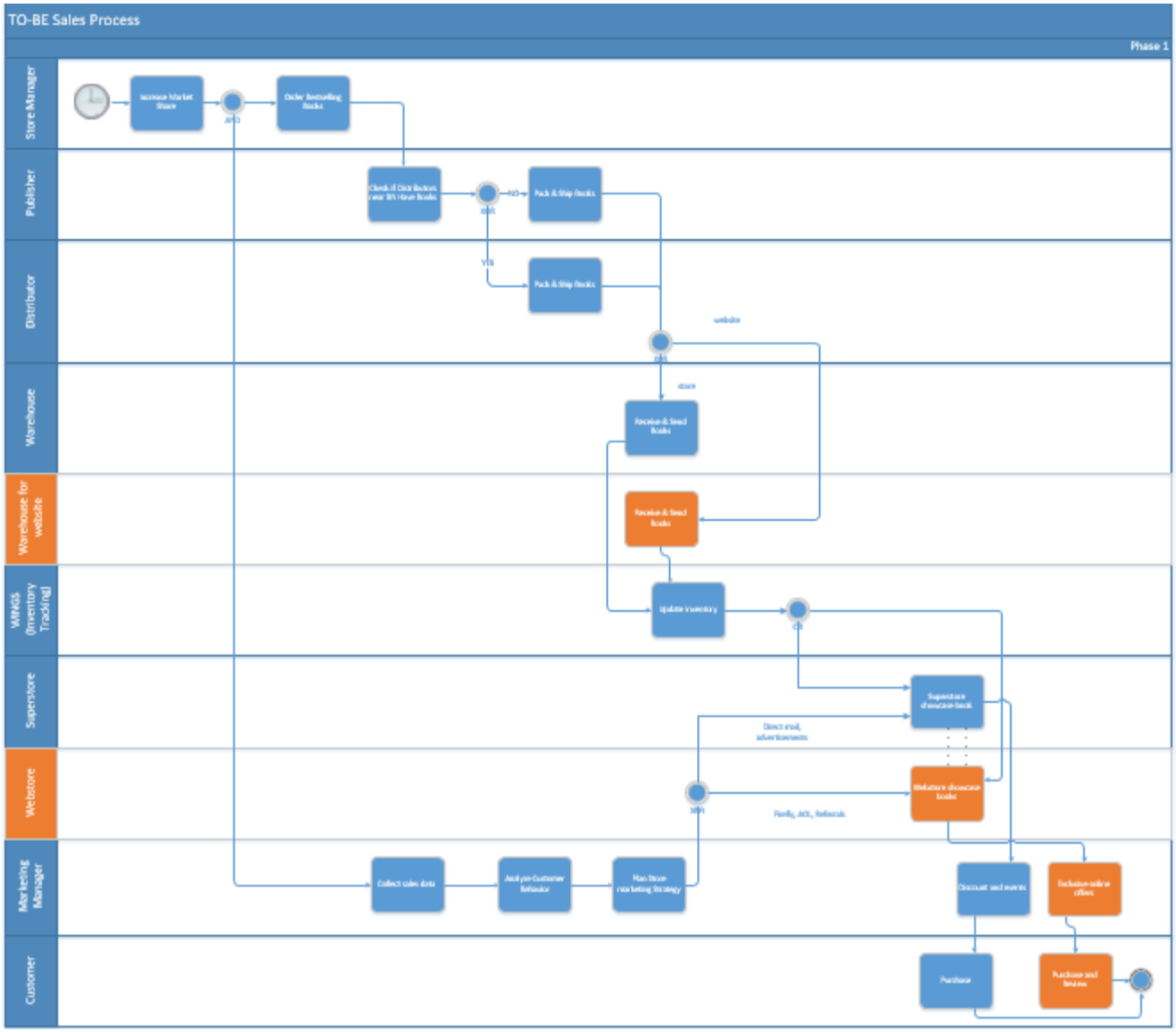

AS-IS Sales Process
Phase 1
Store Manager
Publisher
Distributor
Warehouse
WINGS (Inventory Tracking)
Superstore
Mallstore
Marketing Manager
Customer
Increase Market Share
AND
Order Bestselling Books
Check if Distributors near BN Have Books
XOR
NO
YES
Pack & Ship Books
Pack & Ship Books
Receive & Send Books
Update Inventory
OR
Superstore showcase book
MallStore showcase book
OR
Collect sales data
Analyze Customer Behavior
Plan Store marketing Strategy
Discount
Discount and events
Purchase Books

Figure 2:


TO-BE Sales Process
Phase 1
Store Manager
Publisher
Distributor
Warehouse
Warehouse for website
WMS (Inventory Tracking)
Superstore
Webstore
Marketing Manager
Customer
Increase Market Share
Order Bestselling Books
Check if Distributors near BN have books
Pack & Ship Books
Pack & Ship Books
website
store
Receive & Stock Books
Receive & Stock Books
Update Inventory
Superstore showcase books
Direct mail, advertisements
Webstore showcase books
Flyers, ADS, Referrals
Collect sales data
Analyze Customer Behavior
Plan future marketing Strategy
Discount and events
Exclusive online offers
Purchase
Purchase and Review

2 Database Creation

2.1 Assumptions

The following assumptions has been made for the ease of database creation :

- Table ORDER has been renamed to BOOKORDER and Table STORE has been renamed to STORETYPE.
- In SHIPMENT relational table, 'Shipment Date' and 'Shipment Cost' may or may not be null.
- In BOOKORDER table, Tracking Number is provided only when book is purchased via web.

2.2 Description

In order to build the efficient central database including necessary information about sales process of Barnes & Noble, we review the old sales process and new sales process that is focused on online selling, and then decide to use the modified new sales process because its database is more simplified and efficient. As a result, when creating database of business process of Barnes & Noble, we concentrate on the To-Be sales process that includes the online selling.

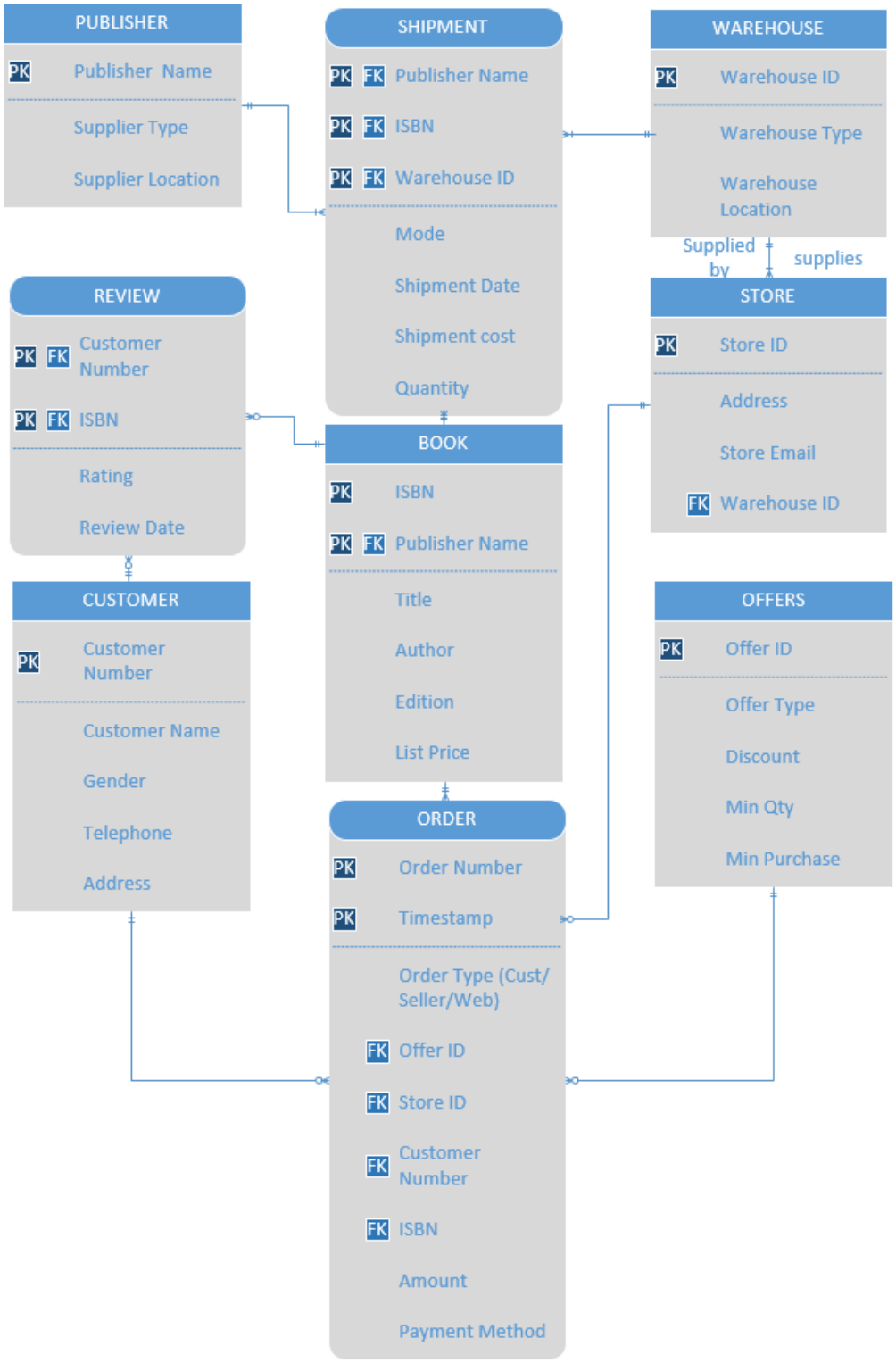
In our view, in order to grasp central and crucial information, information in the database of To-Be process could be divided into three parts. The first part of information is about the supply chain of books. It includes entities such as publisher/distributor, warehouse, store, and shipment. The second part is sales chain of books. It includes entities such as customer, order, and offer. The third part is information of feedbacks of books. It includes review entity. And then we draw ER diagram (Figure 1) that contains the information mentioned before.

Our goal is to create database that improves information exchange among different departments and reduce data redundancy. The goal can be achieved by the database. First, because all the data is imported in the database and can be monitored by computer system, the WINGS, which keep track of the inventory, doesn’t need to update the inventory data about stock, input, and output every time there are books coming or shipping, and to report these inventory data to management. Second, the stores would not need to collect its sales data and

customer information, and report these data to marketing management. Third, marketing managers were faced with more tough problems because they should collect and classify sales data, analyze customers' purchase behavior, and make accurate marketing strategies. by creating the database, it is easier for marketing management to gain brief sales data from other department and thus make marketing strategies with no difficulty.

So, as a result, the advantages in our database design is: 1. Reduce redundancy for relationship; 2. Simplify website related entity; 3. Improve information convey and communication via database.

Figure 3 Entity Relationship (ER) Diagram



2.3 Database Creation

PUBLISHER	
Publisher Name	Primary Key
Publisher Address	Not null
Publisher Email	Not null and unique

SHIPMENT	
Shipment ID	Primary key
Publisher Name	Foreign key to Publisher
ISBN	Foreign key to Book
Warehouse ID	Foreign key to Warehouse
Mode	Two types - Direct shipment & Via Distributor
Shipment Date	
Shipment Cost	
Quantity	Not null

WAREHOUSE	
Warehouse ID	Primary Key
Warehouse Type	Two types - Web & Store
Warehouse Location	Not null

REVIEW	
Customer Number	Primary key, and Foreign key to Customer
ISBN	Primary key, and Foreign key to Book
Rating	Not null
Review Date	Not null

OFFER	
Offer ID	Primary key
Offer Type	Three types – Voucher & Referrals & Promotions
Discount	Not null
Minimum Quantity	Not null
Minimum Purchase	Not null

BOOKORDER	
Order Number	Primary key
Offer ID	Foreign key to Offer
Store ID	Foreign key to Store
Customer Number	Foreign key to Customer
ISBN	Foreign key to Book
Time Stamp	Not null
Amount	Not null
Payment method	Not null
Tracking Number	

CUSTOMER	
Customer Number	Primary key
Customer Name	Not null
Gender	Not null
Telephone	
Address	
Email	

STORETYPE	
Store ID	Primary Key
Store Type	Two types - Webstore & Superstore
Address	Not null
Store Email	Not null and Unique
Warehouse ID	Foreign key to Warehouse

BOOK	
ISBN	Primary key not null
Publisher Name	Foreign key to Publisher not null
Title	Not null
Author	Not null
Edition	
List Price	Not null

2.4 Conclusion

We improve Barnes & Noble's sales business process driven by customers. The redesign of the business process includes reduced number of entities, attributes and relationships, with the purpose of making the process more efficient. More importantly, we divide one complicated entity into two, making the table and entity more easily be understood and connected with each other. Based on these principles, we create a centralized database to store and share data. As a result, we believe that it will significantly improve the efficiency of the business process.

3 Oracle SQL Developer Scripting

3.1 Create Table and Insert Data

In this part, we create tables includes warehouse, offer, customer, publisher, book, shipment, storetype, bookorder and review, and insert data to each of the tables. Our queries are shown below. First we use drop table to make sure there is no other error and we can run it multiple times.

```
drop table REVIEW;
drop table SHIPMENT;
drop table BOOKORDER;
drop table STORETYPE;
drop table BOOK;
drop table PUBLISHER;
drop table CUSTOMER;
drop table OFFER;
drop table WAREHOUSE;
```

```
CREATE TABLE WAREHOUSE
(Warehouse_ID varchar2(7) primary key not null,
Warehouse_Type varchar2(15) not null,
Warehouse_Location varchar2(50));
INSERT INTO WAREHOUSE (WAREHOUSE_ID, WAREHOUSE_TYPE, WAREHOUSE_LOCATION)
VALUES ('W001', 'Store-Warehouse', '2289 Broadway, New York, NY 10024');
```

```
INSERT INTO WAREHOUSE (WAREHOUSE_ID, WAREHOUSE_TYPE, WAREHOUSE_LOCATION)
VALUES ('W002', 'Web-Warehouse', '97 Warren St, New York, NY 10007');

CREATE TABLE OFFER
(Offer_ID varchar2(20) primary key not null,
Offer_Type varchar2(20) not null,
Discount varchar2(20),
Minimum_Quantity number(10)not null,
Minimum_Purchase number(10)not null);
INSERT INTO OFFER (OFFER_ID, OFFER_TYPE, DISCOUNT, MINIMUM_QUANTITY,
MINIMUM_PURCHASE) VALUES ('buy1get1', 'Voucher', '50%', '1', '200');
INSERT INTO OFFER (OFFER_ID, OFFER_TYPE, DISCOUNT, MINIMUM_QUANTITY,
MINIMUM_PURCHASE) VALUES ('buy2get2', 'Voucher', '50%', '2', '300');
INSERT INTO OFFER (OFFER_ID, OFFER_TYPE, DISCOUNT, MINIMUM_QUANTITY,
MINIMUM_PURCHASE) VALUES ('buy2get1', 'Voucher', '33%', '2', '150');
INSERT INTO OFFER (OFFER_ID, OFFER_TYPE, DISCOUNT, MINIMUM_QUANTITY,
MINIMUM_PURCHASE) VALUES ('buy1get2', 'Voucher', '66%', '1', '300');
INSERT INTO OFFER (OFFER_ID, OFFER_TYPE, DISCOUNT, MINIMUM_QUANTITY,
MINIMUM_PURCHASE) VALUES ('storereferral', 'Referral', '20%', '1', '100');
INSERT INTO OFFER (OFFER_ID, OFFER_TYPE, DISCOUNT, MINIMUM_QUANTITY,
MINIMUM_PURCHASE) VALUES ('webreferral', 'Referral', '30%', '1', '75');
INSERT INTO OFFER (OFFER_ID, OFFER_TYPE, DISCOUNT, MINIMUM_QUANTITY,
MINIMUM_PURCHASE) VALUES ('firstpurchase20', 'Promotion', '20%', '1', '50');
INSERT INTO OFFER (OFFER_ID, OFFER_TYPE, DISCOUNT, MINIMUM_QUANTITY,
MINIMUM_PURCHASE) VALUES ('firstpurchase50', 'Promotion', '50%', '3', '200');
INSERT INTO OFFER (OFFER_ID, OFFER_TYPE, DISCOUNT, MINIMUM_QUANTITY,
MINIMUM_PURCHASE) VALUES ('20onselect', 'Promotion', '20%', '1', '75');
INSERT INTO OFFER (OFFER_ID, OFFER_TYPE, DISCOUNT, MINIMUM_QUANTITY,
MINIMUM_PURCHASE) VALUES ('bookmarkfree', 'promotion', '0%', '1', '20');

CREATE TABLE CUSTOMER
(Customer_Number varchar2(10) primary key not null,
```

```
Customer_Name varchar2(30) not null,
Customer_Gender varchar2(10) not null,
Customer_Telephone number(15),
Customer_Address varchar2(50));
INSERT INTO CUSTOMER (CUSTOMER_NUMBER, CUSTOMER_NAME, CUSTOMER_GENDER,
CUSTOMER_TELEPHONE, CUSTOMER_ADDRESS) VALUES ('CNBAN001', 'Ryan Zimmer', 'Male',
'86098765425', '440, asylum ave, hartford-06105');
INSERT INTO CUSTOMER (CUSTOMER_NUMBER, CUSTOMER_NAME, CUSTOMER_GENDER,
CUSTOMER_TELEPHONE, CUSTOMER_ADDRESS) VALUES ('CNBAN002', 'Nancy Lass', 'Female',
'99898765432', '200, farmington avenue, hartford-06105');
INSERT INTO CUSTOMER (CUSTOMER_NUMBER, CUSTOMER_NAME, CUSTOMER_GENDER,
CUSTOMER_TELEPHONE, CUSTOMER_ADDRESS) VALUES ('CNBAN003', 'Michael Bonnenfant',
'Male', '86098765478', '23, main street hartford-06105');
INSERT INTO CUSTOMER (CUSTOMER_NUMBER, CUSTOMER_NAME, CUSTOMER_GENDER,
CUSTOMER_TELEPHONE, CUSTOMER_ADDRESS) VALUES ('CNBAN004', 'Kim Menon', 'Female',
'99878765432', '771, asylum ave, hartford-06105');
INSERT INTO CUSTOMER (CUSTOMER_NUMBER, CUSTOMER_NAME, CUSTOMER_GENDER,
CUSTOMER_TELEPHONE, CUSTOMER_ADDRESS) VALUES ('CNBAN005', 'Billy Hoff', 'Male',
'86098765467', '83, asylum ave, hartford-06105');
INSERT INTO CUSTOMER (CUSTOMER_NUMBER, CUSTOMER_NAME, CUSTOMER_GENDER,
CUSTOMER_TELEPHONE, CUSTOMER_ADDRESS) VALUES ('CNBAN006', 'William S', 'Male',
'99668765432', '512, farmington avenue, hartford-06105');
INSERT INTO CUSTOMER (CUSTOMER_NUMBER, CUSTOMER_NAME, CUSTOMER_GENDER,
CUSTOMER_TELEPHONE, CUSTOMER_ADDRESS) VALUES ('CNBAN007', 'Smith Connor', 'Male',
'99458765432', '66, farmington avenue, hartford-06105');
INSERT INTO CUSTOMER (CUSTOMER_NUMBER, CUSTOMER_NAME, CUSTOMER_GENDER,
CUSTOMER_TELEPHONE, CUSTOMER_ADDRESS) VALUES ('CNBAN008', 'James Long', 'Male',
'86098765445', '21, asylum ave, hartford-06105');
INSERT INTO CUSTOMER (CUSTOMER_NUMBER, CUSTOMER_NAME, CUSTOMER_GENDER,
CUSTOMER_TELEPHONE, CUSTOMER_ADDRESS) VALUES ('CNBAN009', 'Marie Claire', 'Female',
'91998765432', '1020, asylum ave, hartford-06105');
```

```
INSERT INTO CUSTOMER (CUSTOMER_NUMBER, CUSTOMER_NAME, CUSTOMER_GENDER,
CUSTOMER_TELEPHONE, CUSTOMER_ADDRESS) VALUES ('CNBAN010', 'Joan Matthew',
'Female', '86094565432', '725, main street hartford-06105');

CREATE TABLE PUBLISHER
(Publisher_Name varchar2(30) primary key not null,
Publisher_Address varchar2(50) not null,
Publisher_Email varchar2(50)not null unique);
INSERT INTO PUBLISHER (PUBLISHER_NAME, PUBLISHER_ADDRESS, PUBLISHER_EMAIL) VALUES
('Antrim House Books', '21 Goodrich Rd, Simsbury, CT 06070', 'antrim@gmail.com');
INSERT INTO PUBLISHER (PUBLISHER_NAME, PUBLISHER_ADDRESS, PUBLISHER_EMAIL) VALUES
('Tidemark Press Limited', '22 Prestige Park Cir, East Hartford, CT 06108',
'tidemark@gmail.com');
INSERT INTO PUBLISHER (PUBLISHER_NAME, PUBLISHER_ADDRESS, PUBLISHER_EMAIL) VALUES
('Strategic Book Group', 'Durham, CT 06422', 'strategic@gmail.com');
INSERT INTO PUBLISHER (PUBLISHER_NAME, PUBLISHER_ADDRESS, PUBLISHER_EMAIL) VALUES
('For Beginners LLC', '155 Main St #211, Danbury, CT 06810', 'beginners@gmail.com');
INSERT INTO PUBLISHER (PUBLISHER_NAME, PUBLISHER_ADDRESS, PUBLISHER_EMAIL) VALUES
('Yale University Press', '302 Temple St, New Haven, CT 06511', 'yalepress@gmail.com');
INSERT INTO PUBLISHER (PUBLISHER_NAME, PUBLISHER_ADDRESS, PUBLISHER_EMAIL) VALUES
('Nancy Larson Publishers', '302 Temple St, New Haven, CT 06511', 'nancylarson@gmail.com');
INSERT INTO PUBLISHER (PUBLISHER_NAME, PUBLISHER_ADDRESS, PUBLISHER_EMAIL) VALUES
('Unicorn for writers', '17 Church Hill Rd, Redding, CT 06896', 'unicorn@gmail.com');
INSERT INTO PUBLISHER (PUBLISHER_NAME, PUBLISHER_ADDRESS, PUBLISHER_EMAIL) VALUES
('Industrial Press Inc', '32 Haviland St #2C, Norwalk, CT 06854', 'industrialpress@gmail.com');
INSERT INTO PUBLISHER (PUBLISHER_NAME, PUBLISHER_ADDRESS, PUBLISHER_EMAIL) VALUES
('Mirror Books LLC', '175 Atlantic St, Stamford, CT 06901', 'mirrorbooks@gmail.com');
INSERT INTO PUBLISHER (PUBLISHER_NAME, PUBLISHER_ADDRESS, PUBLISHER_EMAIL) VALUES
('Tell Me Press LLC', '98 Mansfield St, New Haven, CT 06511', 'tellmepress@gmail.com');
INSERT INTO PUBLISHER (PUBLISHER_NAME, PUBLISHER_ADDRESS, PUBLISHER_EMAIL) VALUES
('Wiley Group', '68 Greenfield St, Stamford, CT 06712', 'andersonparg@gmail.com');
```

```
CREATE TABLE BOOK
(ISBN varchar2(10) primary key not null,
Publisher_Name varchar2(30) not null,
Title varchar2(50) not null,
Author varchar2(30) not null,
Book_Edition varchar2(10),
List_Price number(10) not null);
INSERT INTO BOOK (ISBN, PUBLISHER_NAME, TITLE, AUTHOR, BOOK_EDITION, LIST_PRICE)
VALUES ('978-0-0314', 'Antrim House Books', 'The Kite Runner', 'Khaled Hosseini', '1.0', '120');
INSERT INTO BOOK (ISBN, PUBLISHER_NAME, TITLE, AUTHOR, BOOK_EDITION, LIST_PRICE)
VALUES ('978-5-0229', 'Tidemark Press Limited', 'The Innocents Aboard', 'Mark Twain', '1.2',
'30');
INSERT INTO BOOK (ISBN, PUBLISHER_NAME, TITLE, AUTHOR, BOOK_EDITION, LIST_PRICE)
VALUES ('338-5-6314', 'Strategic Book Group', 'The Stand', 'Stephen King', '1.0', '40');
INSERT INTO BOOK (ISBN, PUBLISHER_NAME, TITLE, AUTHOR, BOOK_EDITION, LIST_PRICE)
VALUES ('338-3-1256', 'For Beginners LLC', 'The Casual Vacancy', 'J.K.Rowling', '2.3', '200');
INSERT INTO BOOK (ISBN, PUBLISHER_NAME, TITLE, AUTHOR, BOOK_EDITION, LIST_PRICE)
VALUES ('878-1-0356', 'Tell Me Press LLC', 'The Bell Jar', 'Sylvia Plath', '3.7', '100');
INSERT INTO BOOK (ISBN, PUBLISHER_NAME, TITLE, AUTHOR, BOOK_EDITION, LIST_PRICE)
VALUES ('218-0-1234', 'Mirror Books LLC', 'Coraline', 'Neil Gaiman', '2.2', '55');
INSERT INTO BOOK (ISBN, PUBLISHER_NAME, TITLE, AUTHOR, BOOK_EDITION, LIST_PRICE)
VALUES ('911-7-0725', 'Unicorn for writers', 'The Shining', 'Stephen King', '1.7', '80');
INSERT INTO BOOK (ISBN, PUBLISHER_NAME, TITLE, AUTHOR, BOOK_EDITION, LIST_PRICE)
VALUES ('978-6-0314', 'Nancy Larson Publishers', 'Roughing It', 'Mark Twain', '3.1', '95');
INSERT INTO BOOK (ISBN, PUBLISHER_NAME, TITLE, AUTHOR, BOOK_EDITION, LIST_PRICE)
VALUES ('118-0-1114', 'Yale University Press', 'Animal Farm', 'George Orwell', '1.2', '150');
INSERT INTO BOOK (ISBN, PUBLISHER_NAME, TITLE, AUTHOR, BOOK_EDITION, LIST_PRICE)
VALUES ('978-0-9236', 'Industrial Press Inc', 'Emma', 'Jane Austen', '2.5', '125');
INSERT INTO BOOK (ISBN, PUBLISHER_NAME, TITLE, AUTHOR, BOOK_EDITION, LIST_PRICE)
VALUES ('811-7-0726', 'Wiley Group', 'Nature Valley', 'Rick Hartman', '1.5', '177');
```

```
CREATE TABLE SHIPMENT
(Shipment_ID varchar2(10) primary key not null,
Publisher_Name varchar2(30) references publisher (Publisher_Name),
ISBN varchar2(10) references book (ISBN),
Warehouse_ID varchar2(7) references warehouse (Warehouse_ID),
Shipment_Mode varchar2(50),
Shipment_Date varchar2(9),
Shipment_Cost number(7),
Quantity number(7) not null);
INSERT INTO SHIPMENT (SHIPMENT_ID, PUBLISHER_NAME, ISBN, WAREHOUSE_ID,
SHIPMENT_MODE, SHIPMENT_DATE, SHIPMENT_COST, QUANTITY) VALUES ('STR001', 'Antrim
House Books', '978-0-0314', 'W001', 'Direct', '12-dec-13', '0', '1');
INSERT INTO SHIPMENT (SHIPMENT_ID, PUBLISHER_NAME, ISBN, WAREHOUSE_ID,
SHIPMENT_MODE, SHIPMENT_DATE, SHIPMENT_COST, QUANTITY) VALUES ('WEB002',
'Tidemark Press Limited', '978-5-0229', 'W002', 'Distributor', '29-Sep-13', '25', '10');
INSERT INTO SHIPMENT (SHIPMENT_ID, PUBLISHER_NAME, ISBN, WAREHOUSE_ID,
SHIPMENT_MODE, SHIPMENT_DATE, SHIPMENT_COST, QUANTITY) VALUES ('STR002', 'Strategic
Book Group', '338-5-6314', 'W001', 'Direct', '14-Jan-14', '10', '3');
INSERT INTO SHIPMENT (SHIPMENT_ID, PUBLISHER_NAME, ISBN, WAREHOUSE_ID,
SHIPMENT_MODE, SHIPMENT_DATE, SHIPMENT_COST, QUANTITY) VALUES ('WEB003', 'For
Beginners LLC', '338-3-1256', 'W002', 'Distributor', '8-Aug-13', '30', '50');
INSERT INTO SHIPMENT (SHIPMENT_ID, PUBLISHER_NAME, ISBN, WAREHOUSE_ID,
SHIPMENT_MODE, SHIPMENT_DATE, SHIPMENT_COST, QUANTITY) VALUES ('STR003', 'Yale
University Press', '878-1-0356', 'W001', 'Distributor', '3-Mar-13', '20', '50');
INSERT INTO SHIPMENT (SHIPMENT_ID, PUBLISHER_NAME, ISBN, WAREHOUSE_ID,
SHIPMENT_MODE, SHIPMENT_DATE, SHIPMENT_COST, QUANTITY) VALUES ('STR004', 'Nancy
Larson Publishers', '218-0-1234', 'W001', 'Direct', '22-Sep-13', '0', '5');
INSERT INTO SHIPMENT (SHIPMENT_ID, PUBLISHER_NAME, ISBN, WAREHOUSE_ID,
SHIPMENT_MODE, SHIPMENT_DATE, SHIPMENT_COST, QUANTITY) VALUES ('WEB001', 'Unicorn
for writers', '911-7-0725', 'W002', 'Direct', '5-Sep-13', '0', '1');
```

```
INSERT INTO SHIPMENT (SHIPMENT_ID, PUBLISHER_NAME, ISBN, WAREHOUSE_ID,
SHIPMENT_MODE, SHIPMENT_DATE, SHIPMENT_COST, QUANTITY) VALUES ('WEB004',
'Industrial Press Inc', '978-6-0314', 'W002', 'Distributor', '1-Jan-14', '30', '40');
INSERT INTO SHIPMENT (SHIPMENT_ID, PUBLISHER_NAME, ISBN, WAREHOUSE_ID,
SHIPMENT_MODE, SHIPMENT_DATE, SHIPMENT_COST, QUANTITY) VALUES ('STR005', 'Mirror
Books LLC', '118-0-1114', 'W001', 'Distributor', '7-Jun-13', '20', '25');
INSERT INTO SHIPMENT (SHIPMENT_ID, PUBLISHER_NAME, ISBN, WAREHOUSE_ID,
SHIPMENT_MODE, SHIPMENT_DATE, SHIPMENT_COST, QUANTITY) VALUES ('STR006', 'Tell Me
Press LLC', '978-0-9236', 'W001', 'Direct', '1-Apr-13', '10', '7');
INSERT INTO SHIPMENT (SHIPMENT_ID, PUBLISHER_NAME, ISBN, WAREHOUSE_ID,
SHIPMENT_MODE, SHIPMENT_DATE, SHIPMENT_COST, QUANTITY) VALUES ('STR007', 'Wiley
Group', '811-7-0726', 'W001', 'Distributor', '5-Nov-13', '0', '1');

CREATE TABLE STORETYPE
(Store_ID varchar2(10) primary key not null,
Warehouse_ID varchar2(7) references warehouse (Warehouse_ID),
Store_Type varchar2(10),
Store_Email varchar2(50)not null unique,
Address varchar2(50)not null);
INSERT INTO STORETYPE (STORE_ID, WAREHOUSE_ID, STORE_TYPE, STORE_EMAIL, ADDRESS)
VALUES ('BAN101', 'W002', 'webstore', 'ebarnesandnobles@gmail.com', '90 Warren St, New
York, NY 10007');
INSERT INTO STORETYPE (STORE_ID, WAREHOUSE_ID, STORE_TYPE, STORE_EMAIL, ADDRESS)
VALUES ('BAN102', 'W002', 'webstore', 'ebarnesandnobles.com', '267 7th Ave, Brooklyn, NY
11215');
INSERT INTO STORETYPE (STORE_ID, WAREHOUSE_ID, STORE_TYPE, STORE_EMAIL, ADDRESS)
VALUES ('BAN001', 'W001', 'superstore', 'banwesthartford@gmail.com', ' Blue Back Square,
West Hartford, CT 06107');
INSERT INTO STORETYPE (STORE_ID, WAREHOUSE_ID, STORE_TYPE, STORE_EMAIL, ADDRESS)
VALUES ('BAN002', 'W001', 'superstore', 'banfarmington@gmail.com', '1599 SE Rd, Farmington,
CT 06032');
```

```
INSERT INTO STORETYPE (STORE_ID, WAREHOUSE_ID, STORE_TYPE, STORE_EMAIL, ADDRESS)
VALUES ('BAN003', 'W001', 'superstore', 'banglastonbury@gmail.com', '1599 SE Rd, Farmington,
CT 06032');
INSERT INTO STORETYPE (STORE_ID, WAREHOUSE_ID, STORE_TYPE, STORE_EMAIL, ADDRESS)
VALUES ('BAN004', 'W001', 'superstore', 'banbucklandhill@gmail.com', 'Shoppes at Buckland
Hills, Manchester, CT 06040');
INSERT INTO STORETYPE (STORE_ID, WAREHOUSE_ID, STORE_TYPE, STORE_EMAIL, ADDRESS)
VALUES ('BAN005', 'W001', 'superstore', 'banmilford@gmail.com', 'Milford Crossing, Milford, CT
06460');
INSERT INTO STORETYPE (STORE_ID, WAREHOUSE_ID, STORE_TYPE, STORE_EMAIL, ADDRESS)
VALUES ('BAN006', 'W001', 'superstore', 'banwestport@gmail.com', '1076 Post Rd E, Westport,
CT 06880');
INSERT INTO STORETYPE (STORE_ID, WAREHOUSE_ID, STORE_TYPE, STORE_EMAIL, ADDRESS)
VALUES ('BAN007', 'W001', 'superstore', 'bannorthhaven@gmail.com', 'North Haven Crossing,
470 Universal Dr N, CT 06473');
INSERT INTO STORETYPE (STORE_ID, WAREHOUSE_ID, STORE_TYPE, STORE_EMAIL, ADDRESS)
VALUES ('BAN008', 'W001', 'superstore', 'banfarmingtonvalley@gmail.com', 'The Shoppes at
Farmington Valley, Canton, CT 06019');

CREATE TABLE BOOKORDER
(Order_Number varchar2(10) primary key not null,
Offer_ID varchar2(20) references offer (Offer_ID),
ISBN varchar2(20) references book (ISBN) not null,
Store_ID varchar2(10) references STORETYPE (Store_ID),
Customer_Number varchar2(10) references customer (Customer_Number),
Order_Timestamp varchar2(9) not null,
Amount number(7) not null,
Payment_Method varchar2(20) not null,
Tracking_Number varchar2(20));
INSERT INTO BOOKORDER (ORDER_NUMBER, OFFER_ID, STORE_ID, ISBN, CUSTOMER_NUMBER,
ORDER_TIMESTAMP, AMOUNT, PAYMENT_METHOD, TRACKING_NUMBER) VALUES ('ABC345',
'buy1get1', 'BAN006', '978-0-0314', 'CNBAN003', '19-Sep-15', '230', 'creditcard', '');
```

INSERT INTO BOOKORDER (ORDER_NUMBER, OFFER_ID, STORE_ID, ISBN, CUSTOMER_NUMBER, ORDER_TIMESTAMP, AMOUNT, PAYMENT_METHOD, TRACKING_NUMBER) VALUES ('ABC456', '', 'BAN003', '978-5-0229', 'CNBAN004', '2-Apr-15', '320', 'creditcard', '');
INSERT INTO BOOKORDER (ORDER_NUMBER, OFFER_ID, STORE_ID, ISBN, CUSTOMER_NUMBER, ORDER_TIMESTAMP, AMOUNT, PAYMENT_METHOD, TRACKING_NUMBER) VALUES ('ABC567', '', 'BAN001', '338-5-6314', 'CNBAN005', '1-Jun-14', '230', 'cash', '');
INSERT INTO BOOKORDER (ORDER_NUMBER, OFFER_ID, STORE_ID, ISBN, CUSTOMER_NUMBER, ORDER_TIMESTAMP, AMOUNT, PAYMENT_METHOD, TRACKING_NUMBER) VALUES ('XYZ123', 'webreferral', 'BAN101', '338-3-1256', 'CNBAN006', '6-Oct-15', '90', 'cashondelivery', '200012');
INSERT INTO BOOKORDER (ORDER_NUMBER, OFFER_ID, STORE_ID, ISBN, CUSTOMER_NUMBER, ORDER_TIMESTAMP, AMOUNT, PAYMENT_METHOD, TRACKING_NUMBER) VALUES ('XYZ678', 'firstpurchase50', 'BAN101', '878-1-0356', 'CNBAN010', '31-Jul-14', '270', 'paytm', '200056');
INSERT INTO BOOKORDER (ORDER_NUMBER, OFFER_ID, STORE_ID, ISBN, CUSTOMER_NUMBER, ORDER_TIMESTAMP, AMOUNT, PAYMENT_METHOD, TRACKING_NUMBER) VALUES ('XYZ345', 'webreferral', 'BAN101', '218-0-1234', 'CNBAN008', '5-Jan-14', '110', 'paytm', '200034');
INSERT INTO BOOKORDER (ORDER_NUMBER, OFFER_ID, STORE_ID, ISBN, CUSTOMER_NUMBER, ORDER_TIMESTAMP, AMOUNT, PAYMENT_METHOD, TRACKING_NUMBER) VALUES ('XYZ456', '', 'BAN102', '911-7-0725', 'CNBAN009', '1-Jan-15', '330', 'cashondelivery', '200045');
INSERT INTO BOOKORDER (ORDER_NUMBER, OFFER_ID, STORE_ID, ISBN, CUSTOMER_NUMBER, ORDER_TIMESTAMP, AMOUNT, PAYMENT_METHOD, TRACKING_NUMBER) VALUES ('ABC123', 'storereferral', 'BAN001', '978-6-0314', 'CNBAN001', '7-Feb-14', '80', 'creditcard', '');
INSERT INTO BOOKORDER (ORDER_NUMBER, OFFER_ID, STORE_ID, ISBN, CUSTOMER_NUMBER, ORDER_TIMESTAMP, AMOUNT, PAYMENT_METHOD, TRACKING_NUMBER) VALUES ('XYZ234', '20onselect', 'BAN102', '118-0-1114', 'CNBAN007', '14-Nov-15', '90', 'creditcard', '200023');
INSERT INTO BOOKORDER (ORDER_NUMBER, OFFER_ID, STORE_ID, ISBN, CUSTOMER_NUMBER, ORDER_TIMESTAMP, AMOUNT, PAYMENT_METHOD, TRACKING_NUMBER) VALUES ('XYZ567', '20onselect', 'BAN102', '978-0-9236', 'CNBAN007', '4-Mar-15', '190', 'creditcard', '200022');

CREATE TABLE REVIEW

(Customer_Number varchar2(10) references customer (Customer_Number)primary key not null,

ISBN varchar2(10) references book (ISBN) unique not null,

Rating varchar2(5)not null,

```
REVIEW_Date varchar2(9)not null);
INSERT INTO REVIEW (CUSTOMER_NUMBER, ISBN, RATING, REVIEW_DATE) VALUES
('CNBAN001', '978-0-0314', '3/5', '01-mar-15');
INSERT INTO REVIEW (CUSTOMER_NUMBER, ISBN, RATING, REVIEW_DATE) VALUES
('CNBAN002', '978-5-0229', '4.5/5', '30-dec-14');
INSERT INTO REVIEW (CUSTOMER_NUMBER, ISBN, RATING, REVIEW_DATE) VALUES
('CNBAN003', '338-5-6314', '5/5', '02-feb-15');
INSERT INTO REVIEW (CUSTOMER_NUMBER, ISBN, RATING, REVIEW_DATE) VALUES
('CNBAN004', '338-3-1256', '3.7/5', '17-jul-14');
INSERT INTO REVIEW (CUSTOMER_NUMBER, ISBN, RATING, REVIEW_DATE) VALUES
('CNBAN005', '878-1-0356', '2.9/5', '01-jun-14');
INSERT INTO REVIEW (CUSTOMER_NUMBER, ISBN, RATING, REVIEW_DATE) VALUES
('CNBAN006', '218-0-1234', '5/5', '30-mar-14');
INSERT INTO REVIEW (CUSTOMER_NUMBER, ISBN, RATING, REVIEW_DATE) VALUES
('CNBAN007', '911-7-0725', '3.3/5', '22-jan-15');
INSERT INTO REVIEW (CUSTOMER_NUMBER, ISBN, RATING, REVIEW_DATE) VALUES
('CNBAN008', '978-6-0314', '2.6/5', '17-apr-15');
INSERT INTO REVIEW (CUSTOMER_NUMBER, ISBN, RATING, REVIEW_DATE) VALUES
('CNBAN009', '118-0-1114', '4.7/5', '31-may-15');
INSERT INTO REVIEW (CUSTOMER_NUMBER, ISBN, RATING, REVIEW_DATE) VALUES
('CNBAN010', '978-0-9236', '3.9/5', '5-aug-14');
```

3.2 Queries Based on Tables and Data

Chapter 1- Select Statement

Objective: This Query is returning book information using title and author name.

Script:

```
select author ||q'[ is author of ]'|| title
        as "Book Info"
from book;
```

Output:

	Book Info
1	Khaled Hosseini is author of The Kite Runner
2	Mark Twain is author of The Innocents Aboard
3	Stephen King is author of The Stand
4	J.K.Rowling is author of The Casual Vacancy
5	Sylvia Plath is author of The Bell Jar
6	Neil Gaiman is author of Coraline
7	Stephen King is author of The Shining
8	Mark Twain is author of Roughing It
9	George Orwell is author of Animal Farm
10	Jane Austen is author of Emma
11	Rick Hartman is author of Nature Valley

Chapter 2- Like Statement

Objective: This Query is returning book information using title and author name.

Script:

```
select shipment_id, shipment_cost
from shipment
where  quantity>5
and shipment_id like 'STR%'  ;
```

Output:

	SHIPMENT_ID	SHIPMENT_COST
1	STR003	20
2	STR005	20
3	STR006	10

Chapter 3- 'concat' Construct

Objective: Display discounts offered to various customers

Script:

```
select concat('Discount of ', o.discount) || ' was offered to customer ' || c.customer_name
"Trending offers"
from bookorder bo
join customer c on (bo.customer_number = c.customer_number)
join offer o on (bo.offer_id = o.offer_id);
```

Output:

	Trending offers
1	Discount of 20% was offered to customer Ryan Zimmer
2	Discount of 50% was offered to customer Michael Bonnenfant
3	Discount of 30% was offered to customer William S
4	Discount of 20% was offered to customer Smith Connor
5	Discount of 20% was offered to customer Smith Connor
6	Discount of 30% was offered to customer James Long
7	Discount of 50% was offered to customer Joan Matthew

Chapter 4- 'NVL' Construct

Objective: display string 'Offline order' wherever Tracking_id is absent.

Script:

```
select order_number "Order#", customer_number "Customer#", NVL(tracking_number, 'Offline
order') "Tracking#"
from bookorder;
```

Output:

	Order#	Customer#	Tracking#
1	ABC345	CNBAN003	Offline order
2	ABC456	CNBAN004	Offline order
3	ABC567	CNBAN005	Offline order
4	XYZ123	CNBAN006	200012
5	XYZ678	CNBAN010	200056
6	XYZ345	CNBAN008	200034
7	XYZ456	CNBAN009	200045
8	ABC123	CNBAN001	Offline order
9	XYZ234	CNBAN007	200023
10	XYZ567	CNBAN007	200022

Chapter 3 & 4- 'case', 'months_between', and 'sysdate' Construct

Objective: For each order, display its discount, min purchase amount and what class it belongs to (if MP < 100, then returns small purchase; if MP>=100 and MP <200, then returns medium purchase; if MP>=200, then returns large purchase), and how long the discount last.

Script:

```
select order_number, o.discount, o.minimum_purchase Min_Purchase_Amount,
case when o.minimum_purchase < 100 then 'small_purchase'
    when o.minimum_purchase >= 100 and o.minimum_purchase < 200 then 'medium_purchase'
```

```
    when o.minimum_purchase >= 200 then 'large_purchase'
    else 'NA'
    end Min_purchase_Type, round(months_between(sysdate,b.ORDER_TIMESTAMP))
Months_Last
from OFFER o join BOOKORDER b on(o.offer_id=b.offer_id);
```

Output:

	Order#	Discount	Min purchase amount	Offer condition	Months before
1	ABC345	50%	200	large purchase	14
2	XYZ123	30%	75	small purchase	14
3	XYZ678	50%	200	large purchase	28
4	XYZ345	30%	75	small purchase	35
5	ABC123	20%	100	medium purchase	34
6	XYZ234	20%	75	small purchase	12
7	XYZ567	20%	75	small purchase	21

Chapter 3 & 4- 'months_between', and 'case' construct

Objective: Categorize offers based on their minimum purchase conditions, and display number of months passed since order occurred.

Script:

```
select order_number "Order#", o.discount "Discount", o.minimum_purchase "Min purchase
amount",
case when o.minimum_purchase < 100 then 'small purchase'
    when o.minimum_purchase >= 100 and o.minimum_purchase < 200 then 'medium purchase'
    when o.minimum_purchase >= 200 then 'large purchase'
    else 'NA'
    end "Offer condition", round(months_between(sysdate,b.ORDER_TIMESTAMP)) "Months
before"
from offer o join bookorder b on(o.offer_id=b.offer_id);
```

Output:

	Order#	Discount	Min purchase amount	Offer condition	Months before
1	ABC345	50%	200	large purchase	14
2	XYZ123	30%	75	small purchase	14
3	XYZ678	50%	200	large purchase	28
4	XYZ345	30%	75	small purchase	35
5	ABC123	20%	100	medium purchase	34
6	XYZ234	20%	75	small purchase	12
7	XYZ567	20%	75	small purchase	21

Chapter 5- Sum Function

Objective: This Query return total sale made by the store.

Script:

```
select store_id,sum(amount) sales
from bookorder
group by store_id
having sum(amount)>200;
```

Output:

	STORE_ID	SALES
1	BAN102	610
2	BAN003	320
3	BAN006	230
4	BAN001	310
5	BAN101	470

Chapter 6- Group Function

Objective: This query returns average amount spend by male and female.

Script:

```
select c.customer_gender, avg(b.amount)as "average amount"
from bookorder b join customer c using (customer_number) group by c.customer_gender;
```

Output:

	CUSTOMER_GENDER	average amount
1	Male	145.714285714285714285714285714285714286
2	Female	306.666666666666666666666666666666666667

Chapter 7-Sub Query

Objective: This Query Returns Price of the book using author name.

Script:

```
select * from book where list_price=(select list_price from book where author like 'Neil%');
```

Output:

	ISBN	PUBLISHER_NAME	TITLE	AUTHOR	BOOK_EDITION	LIST_PRICE
1	218-0-1234	Mirror Books LLC	Coraline	Neil Gaiman	2.2	55

Chapter 8-Intersect statement

Objective: This Query return customer_number who have recently bought the book from the store.

Script:

select customer_number from bookorder intersect select customer_number from customer;

Output:

	CUSTOMER_NUMBER
1	CNBAN001
2	CNBAN003
3	CNBAN004
4	CNBAN005
5	CNBAN006
6	CNBAN007
7	CNBAN008
8	CNBAN009
9	CNBAN010

Chapter 8- Minus statement

Objective: This Query store_id from where no books were ordered or stores having no sales at all.

Script:

select store_id from storetype minus select store_id from bookorder;

Output:

	STORE_ID
1	BAN002
2	BAN004
3	BAN005
4	BAN007
5	BAN008

Chapter 9- Update

Objective: This query updates list price of the book using author name.

Script:

```
update book set list_price=122 where author like 'Khaled%';
```

Output:

```
1 row updated.
```

Chapter 10- Create Statement (Shown in previous part of "Create Table and Insert Data ")

Chapter 11- Create view and select

Objective: To study which customer made how many purchases online utilizing information from BOOKORDER

Script:

```
-- create view showing #customers who have shopped online
create view online_customers as
select customer_number online_cust, count(*) "No. of purchases"
from bookorder
where tracking_number is not null
group by customer_number;

-- display view data
select * from online_customers;
```

Output:

	ONLINE_CUST	No. of purchases
1	CNBAN009	1
2	CNBAN007	2
3	CNBAN010	1
4	CNBAN008	1
5	CNBAN006	1

Chapter 11- Create view and select

Objective: After viewing the selected table, the view table is no longer useful. So we can drop it.

Script:

drop view online_customers;

Output:

View ONLINE_CUSTOMERS dropped.

Extra DDLs- Add Table Constraint

Objective: In BOOK table, PUBLISHER_NAME is a foreign key that references to the primary key in PUBLISHER table. So we want to add a constraint connecting the two PUBLISHER_NAME in BOOK table and PUBLISHER table.

Script:

ALTER TABLE BOOK

ADD constraint BK_PBLSNM_fk foreign key (PUBLISHER_NAME) references PUBLISHER(PUBLISHER_NAME);

Output:

There is a new constraint created. It gives PUBLISHER_NAME in BOOK table a foreign key that references to PUBLISHER_NAME in PUBLISHER table.

Table BOOK altered.

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME
1	BK_PBLSNM_FK	Foreign_Key	(null)	KUD16101	PUBLISHER	SYS_C00283504
2	SYS_C00283506	Check	"ISBN" IS NOT NULL	(null)	(null)	(null)
3	SYS_C00283507	Check	"PUBLISHER_NAME" IS NOT NULL	(null)	(null)	(null)
4	SYS_C00283508	Check	"TITLE" IS NOT NULL	(null)	(null)	(null)
5	SYS_C00283509	Check	"AUTHOR" IS NOT NULL	(null)	(null)	(null)
6	SYS_C00283510	Check	"LIST_PRICE" IS NOT NULL	(null)	(null)	(null)
7	SYS_C00283511	Primary_Key	(null)	(null)	(null)	(null)

Extra DDLs- Modify Data Type

Objective: Change the data type size from 10 to 30 in ISBN in BOOK table to support content sufficiently.

Script:

```
ALTER TABLE BOOK
MODIFY ISBN VARCHAR2(100);
```

Output:

Data type size of ISBN becomes 30 and is enough to support the content of it.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ISBN	VARCHAR2(10 BYTE)	No	(null)	1	(null)
2	PUBLISHER_NAME	VARCHAR2(30 BYTE)	No	(null)	2	(null)
3	TITLE	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4	AUTHOR	VARCHAR2(30 BYTE)	No	(null)	4	(null)
5	BOOK_EDITION	VARCHAR2(10 BYTE)	Yes	(null)	5	(null)
6	LIST_PRICE	NUMBER(10,0)	No	(null)	6	(null)

```
Table BOOK altered.
```

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ISBN	VARCHAR2(100 BYTE)	No	(null)	1	(null)
2	PUBLISHER_NAME	VARCHAR2(30 BYTE)	No	(null)	2	(null)
3	TITLE	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4	AUTHOR	VARCHAR2(30 BYTE)	No	(null)	4	(null)
5	BOOK_EDITION	VARCHAR2(10 BYTE)	Yes	(null)	5	(null)
6	LIST_PRICE	NUMBER(10,0)	No	(null)	6	(null)

4 Conclusion

By now, we have eventually finished our project. Our project is about a case study from Barnes & Noble. There are three phases we have been through.

In the phase 1, we analyze the business process for Barnes & Noble. To begin with, we read the case in detail and understand that Barnes & Noble was once the leading bookseller in the United Stated and its position was taken by Amazon who sold book online, for which it decided to

redesign its business process and the most important of which is embracing the internet. Then, we figure out the problem with the current process of Barnes & Noble that it is unaware of the change of surrounding environment, have applied inefficient traditional supply chain management method, has a complex process for customers to get a book, lake of efficient information management and feels hard to receive Customers' feedback. Therefore, we find ways to deal with those problems and improve it to be more customer-driven, with the purpose of better data management in the future. For example, we suggest to take the traditional business of Barnes & Noble online and to improve its customer reviewing system.

In the phase 2, we draw an ERD for the business process of Barnes & Noble. The diagram includes entity, attribute, relation, primary key and foreign key. When dealing with the "order" entity, we have been stuck with whether we should divide the "order" into "online order" and "offline order". Then, after brainstorming, we choose to use an attribute named "payment method" to indicate if it's online, because the separation of the "order" entity will make the overall business process more difficult than imagined. Finally, we visualize the ERD in the Visio Studio and make tables in the word, as a preparation for database establishment.

In the phase 3, based on the ERD and tables we have drawn before, we write queries to create relation tables, improve them and input values. However, when we start with the queries, we still find problem for the table. For example, we forget to add the "Review" table and the relation for "book order" and "book". So we make some revisions for the create table function. Also, we make some mistakes for data extracting queries. For example, the ignorance of detail makes some queries that look perfect at first unable to run. In addition, in order to use different functions as many as possible, we don't notice the business need of Barnes & Noble. But when we find out where is wrong, we would discuss it together in group immediately and bring up with ideas to solve those problems.

Although we have finished our project, further steps can be taken to analyze to table we extract from the SQL. For example, we can use the regression model or neutral network in SAS JMP to make predictive analytics for the data and do data visualization. That's more important if we want to generate business insight for the case. We still need to learn more business analytics skills to make us well equipped for our career.

Thanks for our excellent team for contributing to our project. Thank professor Xue Bai and her teaching assistant Shyma for their helpful instructions.