Design & Implementation of a Simplified Bookstore Inventory and Sales System

Ansh Chandnani, Jordan Parton, Sara-Jo Fegley

INFO 210: Database Management Systems, Winter 2020

March 19, 2020

1. **Summary:**
   The proposed inventory and sales system will provide user experience of a relational database. The system provides a realistic experience in which business stakeholders and employees of a bookstore can organize, manage, and track business information. This design will benefit inventory management and customer sales to remain competitive.

2. **Problem Statement:**
   A local bookstore requests a database to assist with business organization, managing and tracking customer and inventory records & relationships.

   a. **Overall Goals:**
   The system will be designed to keep track of the customers, orders, suppliers, and current inventory.

   b. **Context and Importance of the System:**
   For the normal operation of any business, a system needs to be in place to keep track of any products, customers and orders that take place. This allows for the business to fulfill the requests of customers quickly and efficiently. It can also keep track of the inventory of books and the suppliers in order to keep the back end of the business running as efficiently as the front end.

   c. **Scope of Project:**

      i. **IN-Scope:** The system will include specific and limited customer demographic information, the current inventory of books, current orders, and suppliers for the book store.

      ii. **OUT-Scope:** The system will not keep track of expenses for the function of the store, such as bills or repairs. It will not notify when the quantity of a product in the inventory reaches zero. It will not automatically update OrderStatus from ‘InTransit’ to ‘Delivered.’ The system will not deal with the validation of Contact Phone Numbers.

3. Requirements:

a. Data requirements:

i. **Customers:** The system will keep track of the **CustomerID, CustomerFName, CustomerLName, CustomerPhone,** and **CustomerEmail.**

ii. **Suppliers:** The system will track the **SupplierID, SupplierName, SupplierAddress, SupplierPhone,** and **SupplierEmail.**

iii. **Inventory:** The system will track the **SupplierID, BookISBN, QtyOnOrder,** and **OrderDate.**

iv. **Books:** The system will track the **BookISBN, BookName, Genre,** and **QtyOnHand.**

v. **Orders:** This table shall keep track of **OrderID, CustomerID, BookID, OrderDate, OrderPrice,** and **OrderStatus (InTransit or Delivered).**

vi. **Book_Author:** This table shall keep track o**f BookISBN** and **AuthorlD.**

vii. **Author:** This table will track the **AuthorID**, **FirstName**, and **LastName**.

viii. **Discount:** This table will keep track of **DiscountID**, **BookISBN**, **DiscountPercent,** and **DiscountDate.**

b. Business Rules and Logic:

i. **Constraints:** All customers must include both a full first and last name and will be implemented by a **NOT NULL** constraint.

ii. A customer foreign key from the customer table is required prior to creating a purchase invoice.

iii. Every purchase invoice must refer to a single BookID.

iv. **Relationships:** Every book will have an associated key for a supplier. Every Order will have a corresponding key for Book and Customer.

v. **Data Integrity:** All tables will be created to protect the accuracy, validity, and consistency by required data relationships.

c. Sample Output:

i. Display customers with outstanding orders.

ii. Display all books in the database and their quantities.

iii. Display all books in a certain genre name.

iv. Display all suppliers.

v. Display all books that are marked at a discounted percentage.

d. **Other Assumptions:**

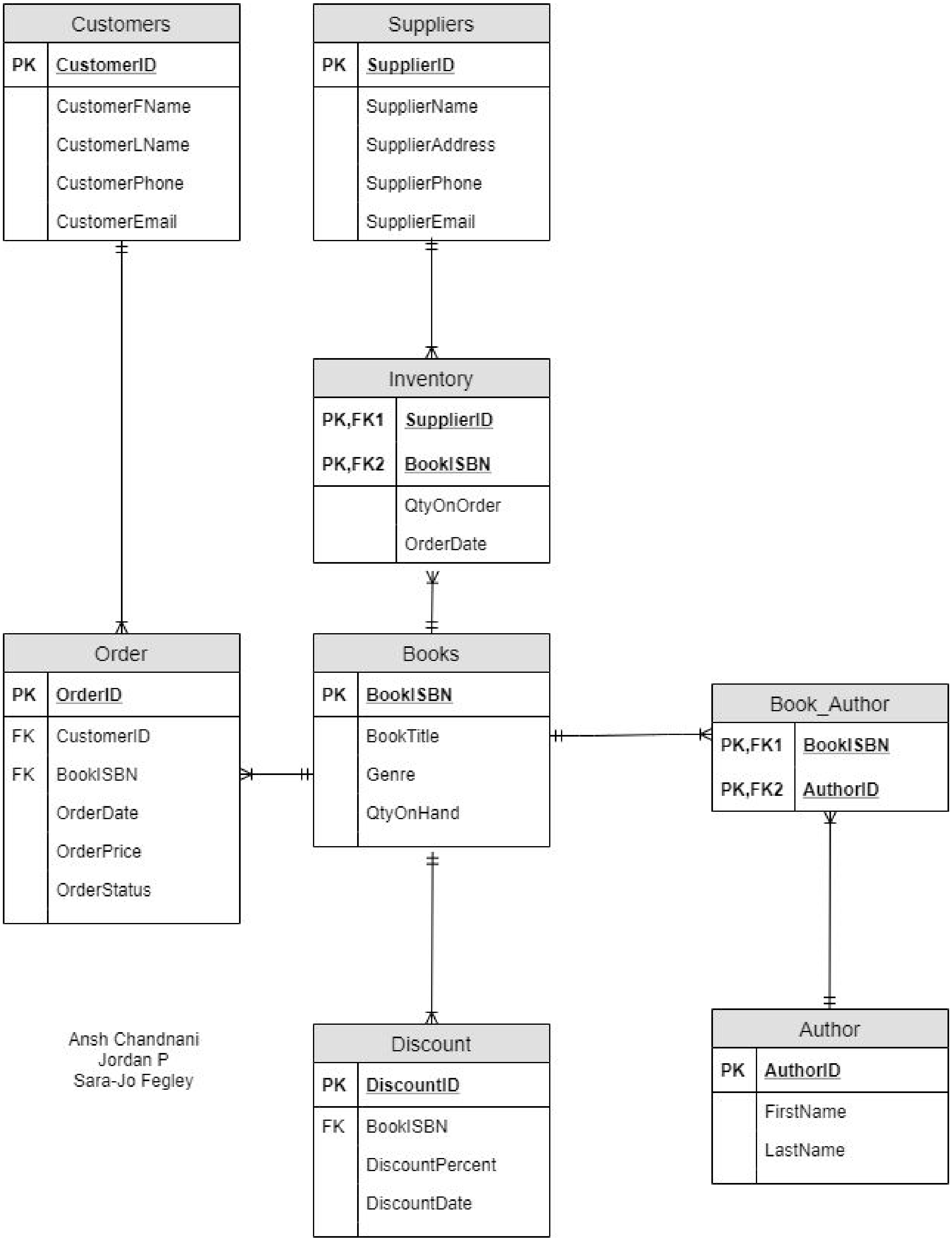
   i. **Application:** The system will be used for the small local bookstore.

   ii. **Customer Information and Privacy:** Personally identifiable information of customers will be identified and safeguarded by the business' standard operating procedures.

   iii. **Electronic Security:** Employee and management structure will be considered when control access is created for users.

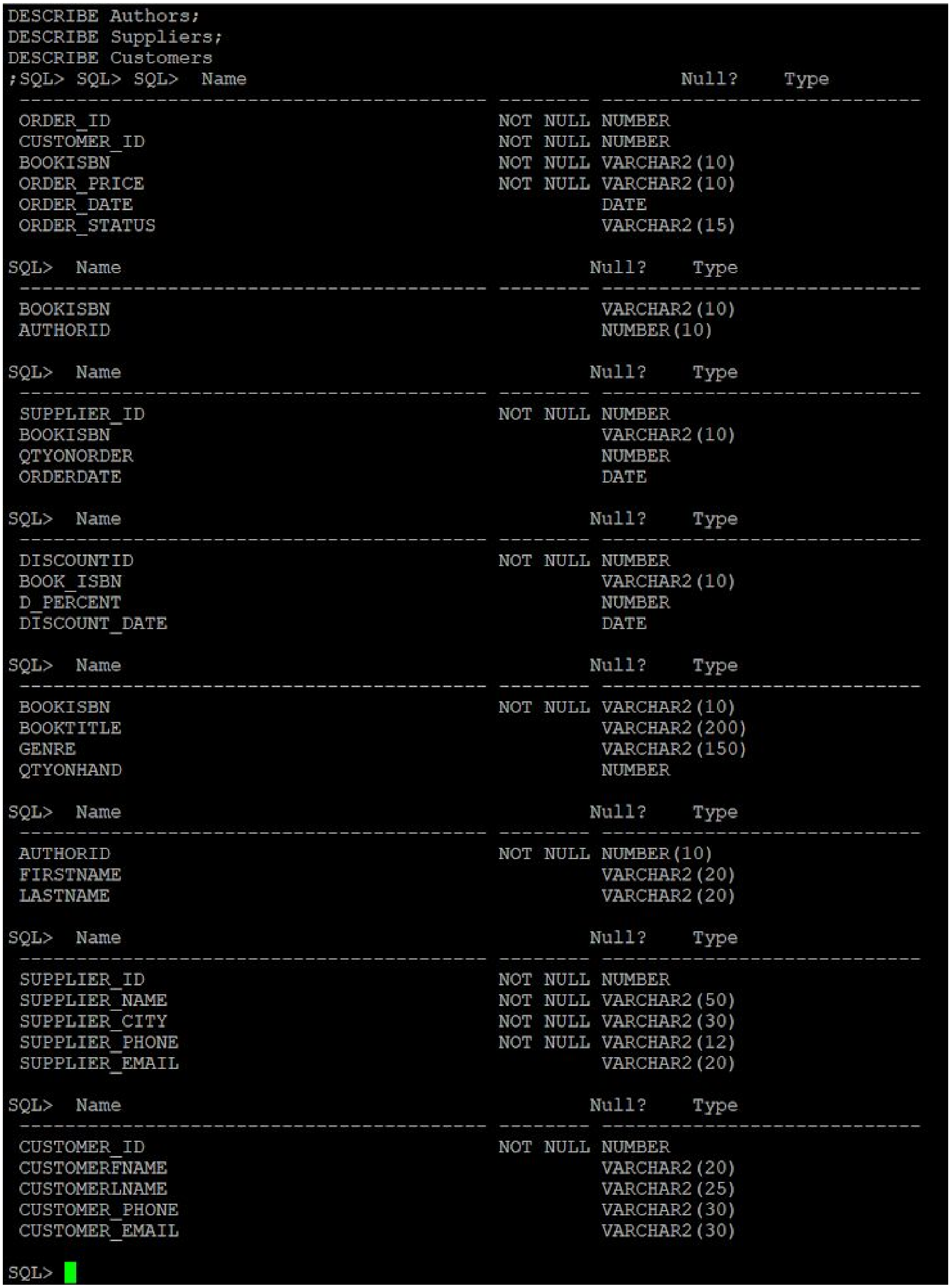
   iv. **Platforms:** Entity Relationship Diagrams will be created using draw.io. Oracle will be used for database creation. The final application will be PC-based running on Linux.

4. Conceptual Design:



5. **Relational Schema:**
    a. **Customers: CustomerID,** CustomerFName, CustomerLName, CustomerPhone, CustomerEmail
    b. **Books: BooksISBN,** BookTitle, Genre, QtyOnHand
    c. **Author: AuthorID,** FirstName, LastName
    d. **Suppliers: SupplierID,** SupplierName, SupplierAddress, SupplierPhone, SupplierEmail
    e. **Inventory: BookISBN REFERENCES Books(BookISBN), SupplierID REFERENCES Suppliers(SupplierID),** QtyOnOrder, OrderDate
    f. **Discount: DiscountID, BookISBN REFERENCES Books(BookISBN),** DiscountPercent, DiscountDate
    g. **Book_Author: BookISBN REFERENCES Books(BookISBN), AuthorID REFERENCES Author(AuthorID)**
    h. **Orders: OrderID, CustomerID REFERENCES Customer(CustomerID), BookISBN REFERENCES Books(BookISBN),** OrderDate, OrderPrice, OrderStatus

6. Data Dictionary:



```
DESCRIBE Authors;
DESCRIBE Suppliers;
DESCRIBE Customers
;SQL> SQL> SQL>  Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ORDER_ID                                  NOT NULL NUMBER
 CUSTOMER_ID                               NOT NULL NUMBER
 BOOKISBN                                  NOT NULL VARCHAR2(10)
 ORDER_PRICE                               NOT NULL VARCHAR2(10)
 ORDER_DATE                                         DATE
 ORDER_STATUS                                       VARCHAR2(15)

SQL>  Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOKISBN                                           VARCHAR2(10)
 AUTHORID                                           NUMBER(10)

SQL>  Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SUPPLIER_ID                               NOT NULL NUMBER
 BOOKISBN                                           VARCHAR2(10)
 QTYONORDER                                         NUMBER
 ORDERDATE                                          DATE

SQL>  Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DISCOUNTID                                NOT NULL NUMBER
 BOOK_ISBN                                          VARCHAR2(10)
 D_PERCENT                                          NUMBER
 DISCOUNT_DATE                                      DATE

SQL>  Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BOOKISBN                                  NOT NULL VARCHAR2(10)
 BOOKTITLE                                          VARCHAR2(200)
 GENRE                                              VARCHAR2(150)
 QTYONHAND                                          NUMBER

SQL>  Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 AUTHORID                                  NOT NULL NUMBER(10)
 FIRSTNAME                                          VARCHAR2(20)
 LASTNAME                                           VARCHAR2(20)

SQL>  Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SUPPLIER_ID                               NOT NULL NUMBER
 SUPPLIER_NAME                             NOT NULL VARCHAR2(50)
 SUPPLIER_CITY                             NOT NULL VARCHAR2(30)
 SUPPLIER_PHONE                            NOT NULL VARCHAR2(12)
 SUPPLIER_EMAIL                                     VARCHAR2(20)

SQL>  Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL NUMBER
 CUSTOMERFNAME                                      VARCHAR2(20)
 CUSTOMERLNAME                                      VARCHAR2(25)
 CUSTOMER_PHONE                                     VARCHAR2(30)
 CUSTOMER_EMAIL                                     VARCHAR2(30)

SQL>
```

7. **Data (Only the first five INSERT statements were included in this report. See DBMS script for all INSERT statements):**

   a. **Customers**

```
CREATE TABLE Customers (
        customer_id NUMBER PRIMARY KEY,
        customerfname VARCHAR(20),
        customerlname VARCHAR(25),
        customer_phone VARCHAR(30),
        customer_email VARCHAR(30)
        );

INSERT INTO Customers
VALUES(1,'Kwai','Yu',2125557818,'yu@gmail.com');
INSERT INTO Customers
VALUES(2,'Paul','Henriot','26.47.1555','henriot@gmail.com');
INSERT INTO Customers VALUES(3,'Daniel','Da Cunha','+33 1 46 62
7555','dacunha@gmail.com');
INSERT INTO Customers
VALUES(4,'Julie','Young',6265557265,'young@gmail.com');
INSERT INTO Customers
VALUES(5,'Julie','Brown',6505551386,'brown@gmail.com');
```

   b. **Suppliers**

```
CREATE TABLE Suppliers (
        Supplier_ID NUMBER PRIMARY KEY NOT NULL,
        Supplier_Name VARCHAR2(50) NOT NULL,
        Supplier_City VARCHAR2(30) NOT NULL,
        Supplier_Phone VARCHAR2(12) NOT NULL,
        Suppplier_Email VARCHAR2(20) NULL
        );
INSERT INTO Suppliers VALUES(1,'ROYAL BOOK CORP','New York
City','123-456-7890','books@rbc.com');
INSERT INTO Suppliers VALUES(2,'SANTA BOOKS USA
INC','Chicago','123-456-7891','books@santab.com');
INSERT INTO Suppliers VALUES(3,'JIM BEAM BRANDS
CO','Detroit','123-456-7892','books@jbb.com');
INSERT INTO Suppliers VALUES(4,'HEAVEN HILL PUBLISHERS
INC','Washington D.C.','123-456-7893','books@hhp.com');
```

```
INSERT INTO Suppliers VALUES(5,'REPUBLIC NATIONAL
DISTRIBUTING CO','Seattle','123-456-7894','books@rnd.com');
```

c. Inventory

```
CREATE TABLE Inventory (
        Supplier_ID NUMBER PRIMARY KEY,
        BookISBN VARCHAR(10),
        QtyOnOrder NUMBER,
        OrderDate DATE,
        FOREIGN KEY (Supplier_ID) REFERENCES
        Suppliers(Supplier_ID),
        FOREIGN KEY (BookISBN) REFERENCES Books(BookISBN)
        );

INSERT INTO Inventory VALUES (1, 9780102848, 10, date
'2015-08-09');
INSERT INTO Inventory VALUES (2, 9780102877, 15, date
'2007-07-12');
INSERT INTO Inventory VALUES (6, 9780102856, 5, date
'2008-01-15');
INSERT INTO Inventory VALUES (10, 9780102855, 10, date
'2003-01-11');
INSERT INTO Inventory VALUES (18, 9780102884, 15, date
'2000-01-01');
```

d. Orders

```
CREATE TABLE Orders (
        Order_ID NUMBER PRIMARY KEY NOT NULL,
        Customer_ID NUMBER NOT NULL,
        BookISBN VARCHAR2(10) NOT NULL,
        Order_Price VARCHAR2(10) NOT NULL,
        Order_Date DATE,
        Order_Status VARCHAR2(15),
        FOREIGN KEY (Customer_ID) REFERENCES
        Customers(customer_ID),
        FOREIGN KEY (BookISBN) REFERENCES Books(BookISBN)
        );
```

```
INSERT INTO Orders VALUES(1,1,9780102846,'95.7',date
'2003-2-24','Shipped');
INSERT INTO Orders
VALUES(2,2,9780102847,'81.35',date'2003-5-7','Shipped');
INSERT INTO Orders VALUES(3,3,9780102848,'94.74',date
'2003-7-1','Shipped');
INSERT INTO Orders VALUES(4,4,9780102849,'83.26',date
'2003-8-25','Shipped');
INSERT INTO Orders VALUES(5,5,9780102850,'100',date
'2003-10-10','Shipped');
```

e. **Books**

```
CREATE TABLE Books (
        BookISBN VARCHAR2(10) PRIMARY KEY,
        BookTitle VARCHAR2(200),
        Genre VARCHAR2(150),
        QtyOnHand NUMBER
        );

INSERT INTO Books VALUES (9780102846,'The Great Escape','Art
Books',6);
INSERT INTO Books VALUES (9780102847,'Underbelly : The
Gangland War','Art Books',4);
INSERT INTO Books VALUES (9780102848,'Oxford Guide to Plain
English','Art Books',9);
INSERT INTO Books VALUES (9780102849,'Get Talking and Keep
Talking Portuguese Total Audio Course ','Art Books',7);
INSERT INTO Books VALUES (9780102850,'The Truthful Art : Data,
Charts, and Maps for Communication','Art Books',1);
```

f. **Books_Authors**

```
DROP TABLE Books_Authors;
        CREATE TABLE Books_Authors (
        BookISBN CONSTRAINT CK_BookISBN REFERENCES
        Books(BookISBN),
        AuthorID CONSTRAINT CK_AuthorID REFERENCES
        Authors(AuthorID)
        );
```

```
INSERT INTO Books_Authors (BookISBN, AuthorID)
SELECT Books.BookISBN, Authors.AuthorID
FROM Books, Authors;
```

g. Authors

```
CREATE TABLE Authors (
        AuthorID NUMBER(10) PRIMARY KEY,
        FirstName VARCHAR(20) NULL,
        LastName VARCHAR(20) NULL
        );

INSERT INTO Authors VALUES (1,'John','Douglas');
INSERT INTO Authors VALUES (2,'Mark','Olshaker');
INSERT INTO Authors VALUES (3,'Cecelia','Mecca');
INSERT INTO Authors VALUES (4,'Ana Maria','Spagna');
INSERT INTO Authors VALUES (5,'Mark','Vanhoenacker');
```

h. Discount

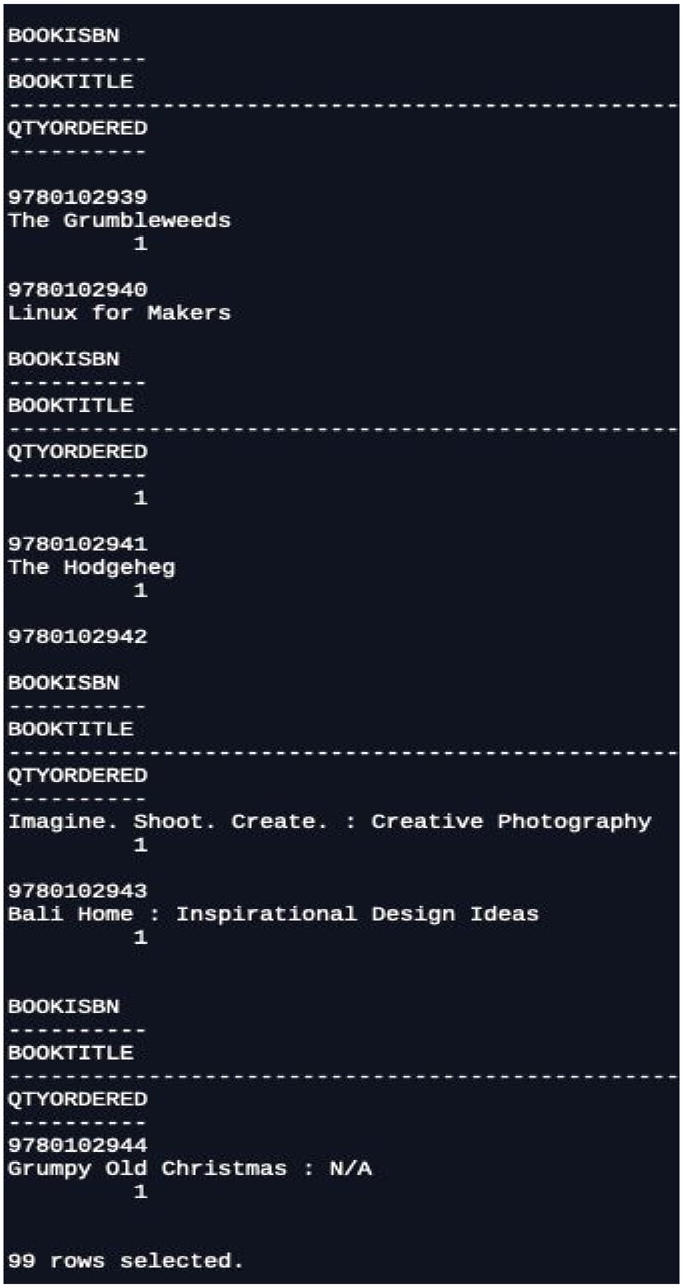
```
DROP TABLE Discount;
CREATE TABLE Discount (
        DiscountID NUMBER PRIMARY KEY,
        Book_ISBN VARCHAR2(10),
        D_Percent NUMBER,
        Discount_Date DATE,
        FOREIGN KEY (Book_ISBN) REFERENCES Books(BookISBN)
        );

INSERT INTO Discount VALUES (1, 9780102846, 15, date
'2003-08-09');
INSERT INTO Discount VALUES (2, 9780102847, 15, date
'2003-10-31');
INSERT INTO Discount VALUES (3, 9780102848, 75, date
'2003-09-10');
INSERT INTO Discount VALUES (4, 9780102849, 15, date
'2003-08-25');
INSERT INTO Discount VALUES (5, 9780102850, 20, date
'2003-01-16');
```

8. Data Queries:

a. Queries by Ansh Chandnani

i. Query 1: Display Book names and ISBNs along with the Quantity Ordered of each Book

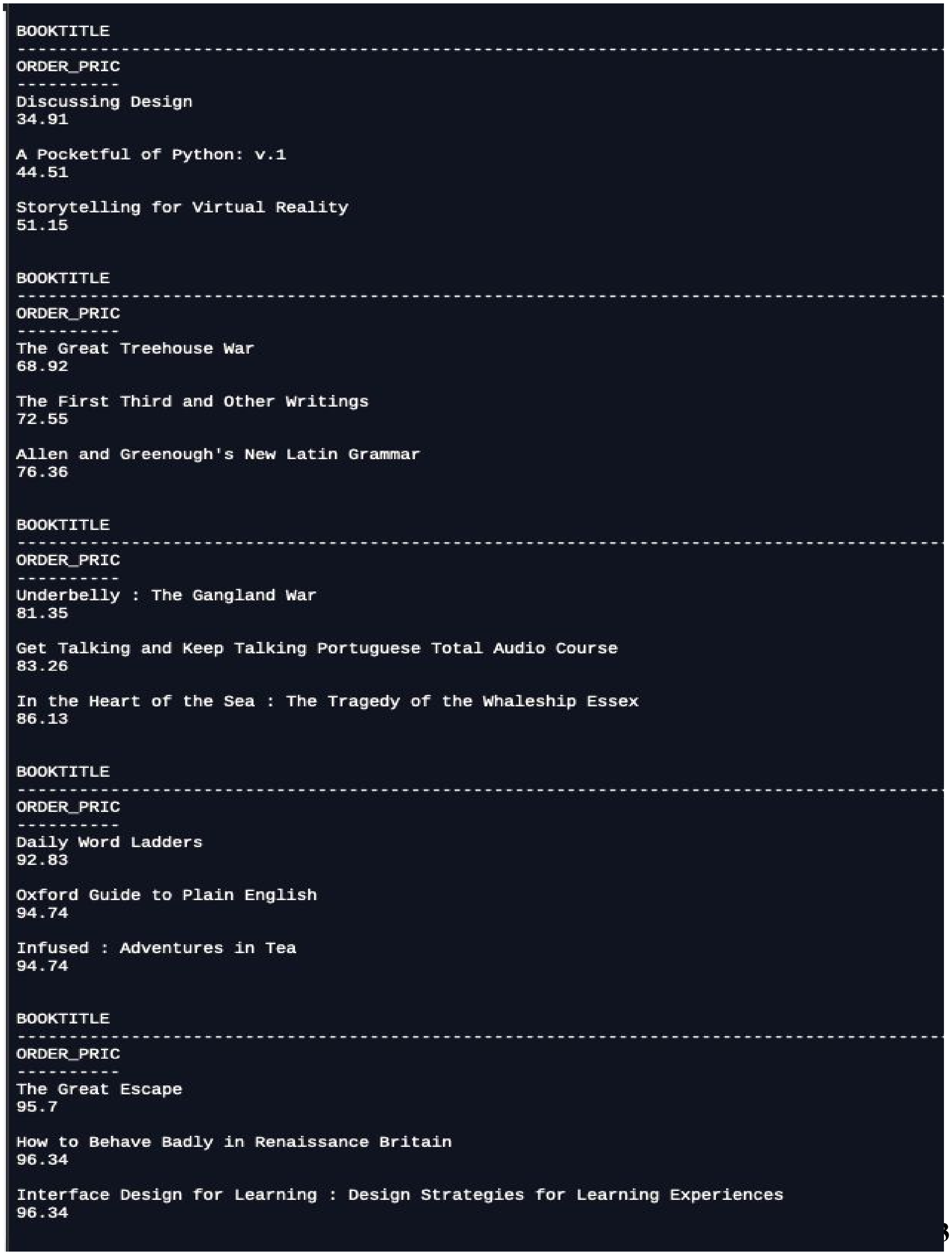
```
SELECT Books.BookISBN, Books.BookTitle, COUNT(Orders.Order_ID)
AS QtyOrdered
FROM Books, Orders
WHERE Books.BookISBN = Orders.BookISBN
GROUP BY Books.BookISBN, Books.BookTitle;
```



```
BOOKISBN
----------
BOOKTITLE
------------------------------------------------
QTYORDERED
----------

9780102939
The Grumbleweeds
         1

9780102940
Linux for Makers

BOOKISBN
----------
BOOKTITLE
------------------------------------------------
QTYORDERED
----------
         1

9780102941
The Hodgeheg
         1

9780102942

BOOKISBN
----------
BOOKTITLE
------------------------------------------------
QTYORDERED
----------
Imagine. Shoot. Create. : Creative Photography
         1

9780102943
Bali Home : Inspirational Design Ideas
         1


BOOKISBN
----------
BOOKTITLE
------------------------------------------------
QTYORDERED
----------
9780102944
Grumpy Old Christmas : N/A
         1


99 rows selected.
```

ii. Query 2: Display Book names from Orders with Price less than 100

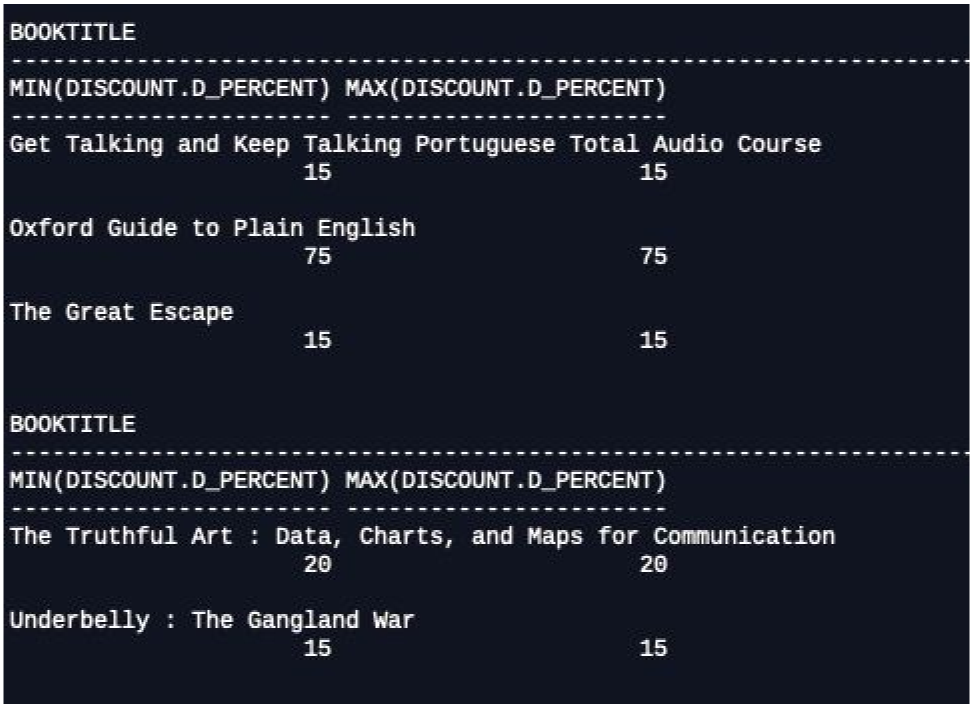
```
SELECT Books.BookTitle, Orders.Order_Price
FROM Books, Orders
WHERE Books.BookISBN = Orders.BookISBN AND
Orders.Order_Price < 100
ORDER BY Orders.Order_Price ASC;
```



```
BOOKTITLE
-------------------------------------------------------------------------------------------
ORDER_PRIC
----------
Discussing Design
34.91

A Pocketful of Python: v.1
44.51

Storytelling for Virtual Reality
51.15


BOOKTITLE
-------------------------------------------------------------------------------------------
ORDER_PRIC
----------
The Great Treehouse War
68.92

The First Third and Other Writings
72.55

Allen and Greenough's New Latin Grammar
76.36


BOOKTITLE
-------------------------------------------------------------------------------------------
ORDER_PRIC
----------
Underbelly : The Gangland War
81.35

Get Talking and Keep Talking Portuguese Total Audio Course
83.26

In the Heart of the Sea : The Tragedy of the Whaleship Essex
86.13


BOOKTITLE
-------------------------------------------------------------------------------------------
ORDER_PRIC
----------
Daily Word Ladders
92.83

Oxford Guide to Plain English
94.74

Infused : Adventures in Tea
94.74


BOOKTITLE
-------------------------------------------------------------------------------------------
ORDER_PRIC
----------
The Great Escape
95.7

How to Behave Badly in Renaissance Britain
96.34

Interface Design for Learning : Design Strategies for Learning Experiences
96.34
```

iii. **Query 3: Display the Book Title along with the maximum and minimum discount percentage**

```
SELECT DISTINCT Books.BookTitle, MIN(Discount.D_Percent),
MAX(Discount.D_Percent)
FROM Books, Discount
WHERE Books.BookISBN = Discount.Book_ISBN
GROUP BY Books.BookTitle;
```

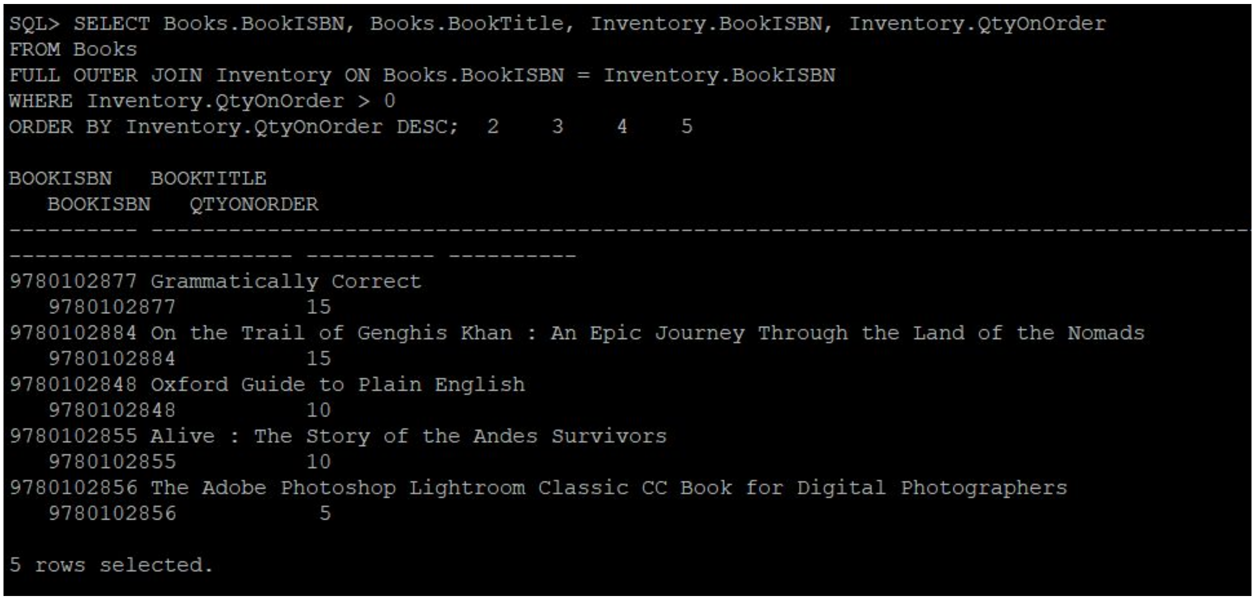


```
BOOKTITLE
---------------------------------------------------------------------
MIN(DISCOUNT.D_PERCENT) MAX(DISCOUNT.D_PERCENT)
----------------------- -----------------------
Get Talking and Keep Talking Portuguese Total Audio Course
                     15                      15

Oxford Guide to Plain English
                     75                      75

The Great Escape
                     15                      15


BOOKTITLE
---------------------------------------------------------------------
MIN(DISCOUNT.D_PERCENT) MAX(DISCOUNT.D_PERCENT)
----------------------- -----------------------
The Truthful Art : Data, Charts, and Maps for Communication
                     20                      20

Underbelly : The Gangland War
                     15                      15
```

b. **Queries by Jordan Parton**

i. **Query 1: Select all book ISBNs and Title where the QtyOnOrder was greater than 0**

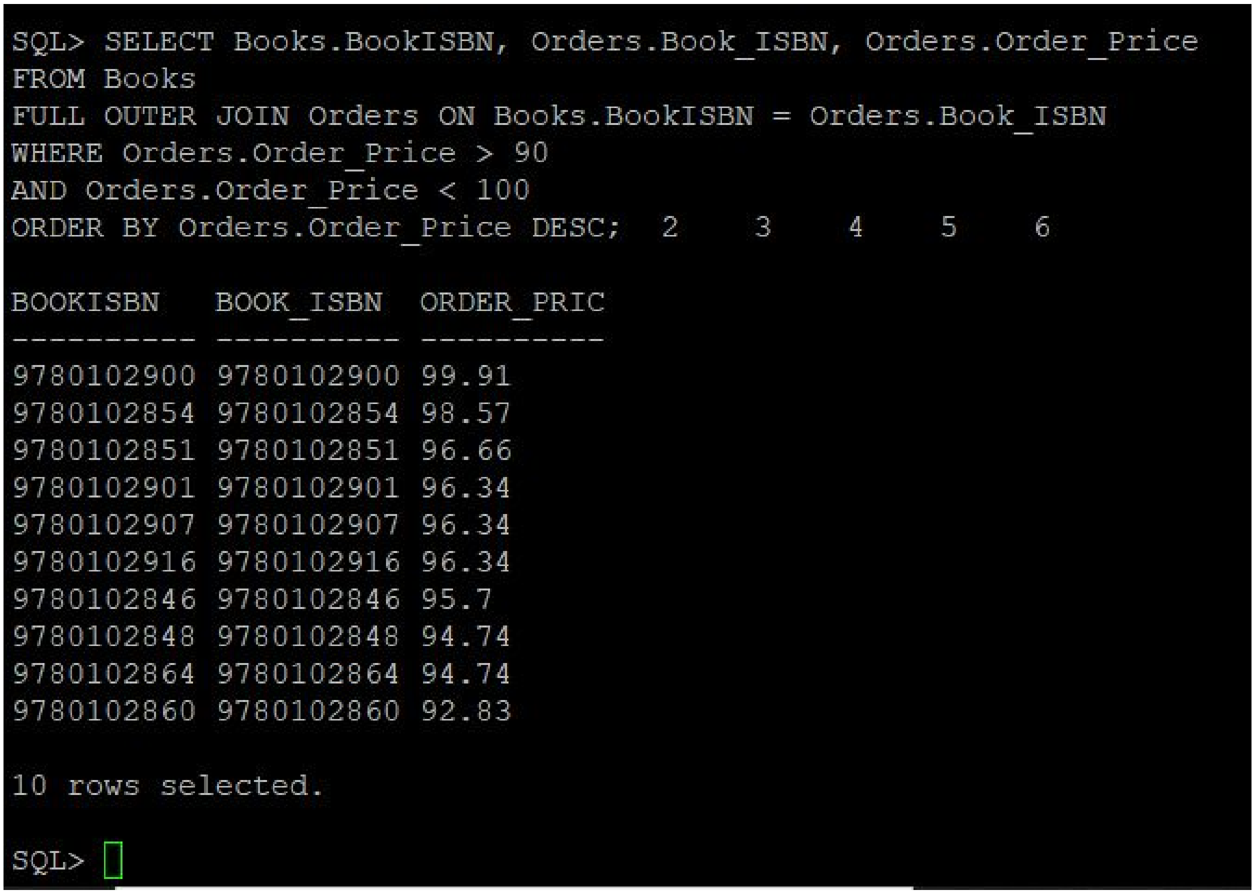
```
SELECT Books.BookISBN, Books.BookTitle,
Inventory.BookISBN, Inventory.QtyOnOrder
FROM Books
FULL OUTER JOIN Inventory ON Books.BookISBN =
Inventory.BookISBN
WHERE Inventory.QtyOnOrder > 0
ORDER BY Inventory.QtyOnOrder DESC;
```



```
SQL> SELECT Books.BookISBN, Books.BookTitle, Inventory.BookISBN, Inventory.QtyOnOrder
FROM Books
FULL OUTER JOIN Inventory ON Books.BookISBN = Inventory.BookISBN
WHERE Inventory.QtyOnOrder > 0
ORDER BY Inventory.QtyOnOrder DESC;  2    3    4    5

BOOKISBN   BOOKTITLE
   BOOKISBN   QTYONORDER
---------- ----------------------------------------------------------------------------------
--------------------- ---------- ----------
9780102877 Grammatically Correct
   9780102877         15
9780102884 On the Trail of Genghis Khan : An Epic Journey Through the Land of the Nomads
   9780102884         15
9780102848 Oxford Guide to Plain English
   9780102848         10
9780102855 Alive : The Story of the Andes Survivors
   9780102855         10
9780102856 The Adobe Photoshop Lightroom Classic CC Book for Digital Photographers
   9780102856          5

5 rows selected.
```

ii. **Query 2: Select the book ISBN of all books in which the order price is greater than 90 but less than 100**

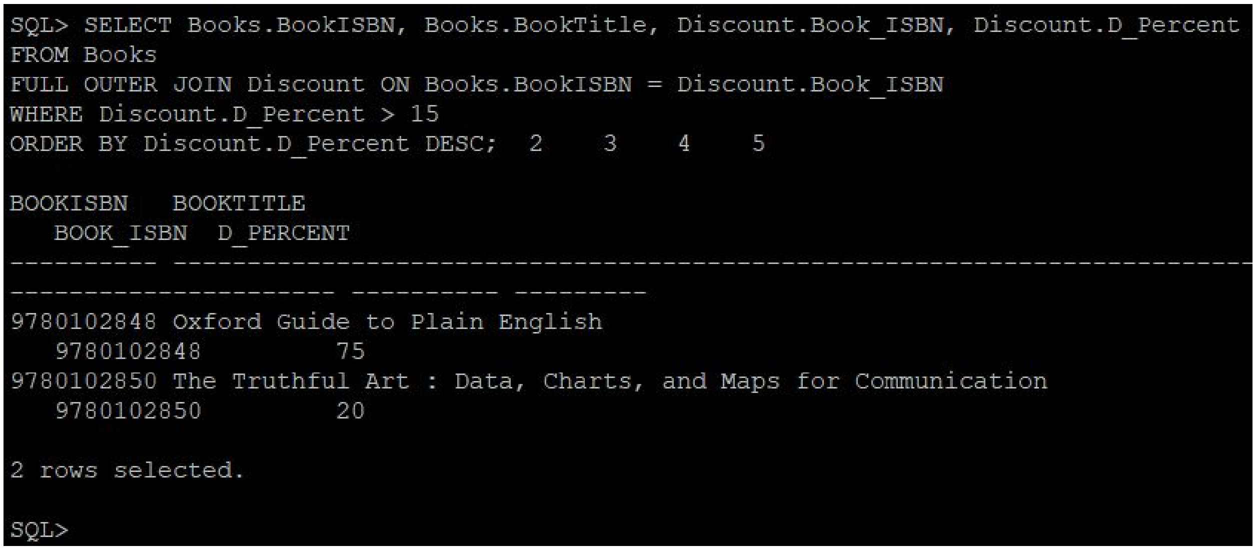
```
SELECT Books.BookISBN, Orders.Book_ISBN,
Orders.Order_Price
FROM Books
FULL OUTER JOIN Orders ON Books.BookISBN =
Orders.Book_ISBN
WHERE Orders.Order_Price > 90
AND Orders.Order_Price < 100
ORDER BY Orders.Order_Price DESC;
```



```
SQL> SELECT Books.BookISBN, Orders.Book_ISBN, Orders.Order_Price
FROM Books
FULL OUTER JOIN Orders ON Books.BookISBN = Orders.Book_ISBN
WHERE Orders.Order_Price > 90
AND Orders.Order_Price < 100
ORDER BY Orders.Order_Price DESC;  2    3    4    5    6

BOOKISBN   BOOK_ISBN  ORDER_PRIC
---------- ---------- ----------
9780102900 9780102900 99.91
9780102854 9780102854 98.57
9780102851 9780102851 96.66
9780102901 9780102901 96.34
9780102907 9780102907 96.34
9780102916 9780102916 96.34
9780102846 9780102846 95.7
9780102848 9780102848 94.74
9780102864 9780102864 94.74
9780102860 9780102860 92.83

10 rows selected.

SQL>
```

iii. Query 3: Find all books that have a discount larger than 15 percent

```
SELECT Books.BookISBN, Books.BookTitle,
Discount.Book_ISBN, Discount.D_Percent
FROM Books
FULL OUTER JOIN Discount ON Books.BookISBN =
Discount.Book_ISBN
WHERE Discount.D_Percent > 15
ORDER BY Discount.D_Percent DESC;
```

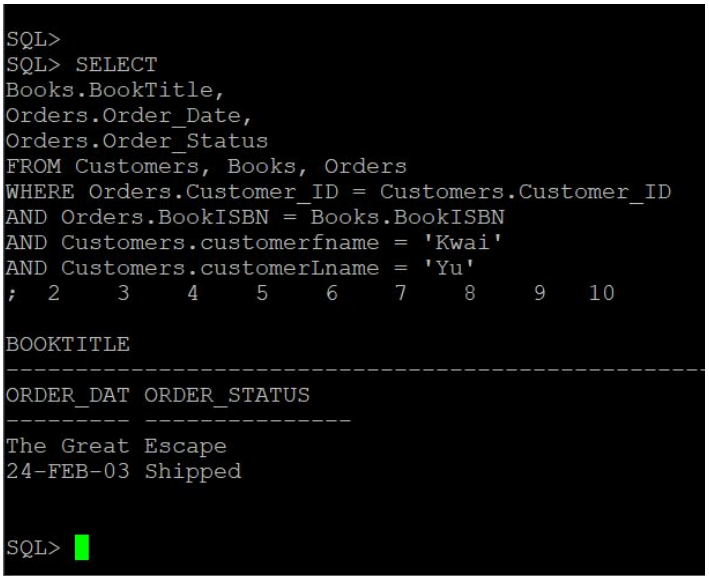


```
SQL> SELECT Books.BookISBN, Books.BookTitle, Discount.Book_ISBN, Discount.D_Percent
FROM Books
FULL OUTER JOIN Discount ON Books.BookISBN = Discount.Book_ISBN
WHERE Discount.D_Percent > 15
ORDER BY Discount.D_Percent DESC;  2    3    4    5

BOOKISBN   BOOKTITLE
   BOOK_ISBN  D_PERCENT
---------- ------------------------------------------------------------------------------
---------------------- ---------- ---------
9780102848 Oxford Guide to Plain English
   9780102848         75
9780102850 The Truthful Art : Data, Charts, and Maps for Communication
   9780102850         20

2 rows selected.

SQL>
```

c. **Queries by Sara-Jo Fegley**

i. **Query 1: All orders made by customer 'Kwai Yu'**

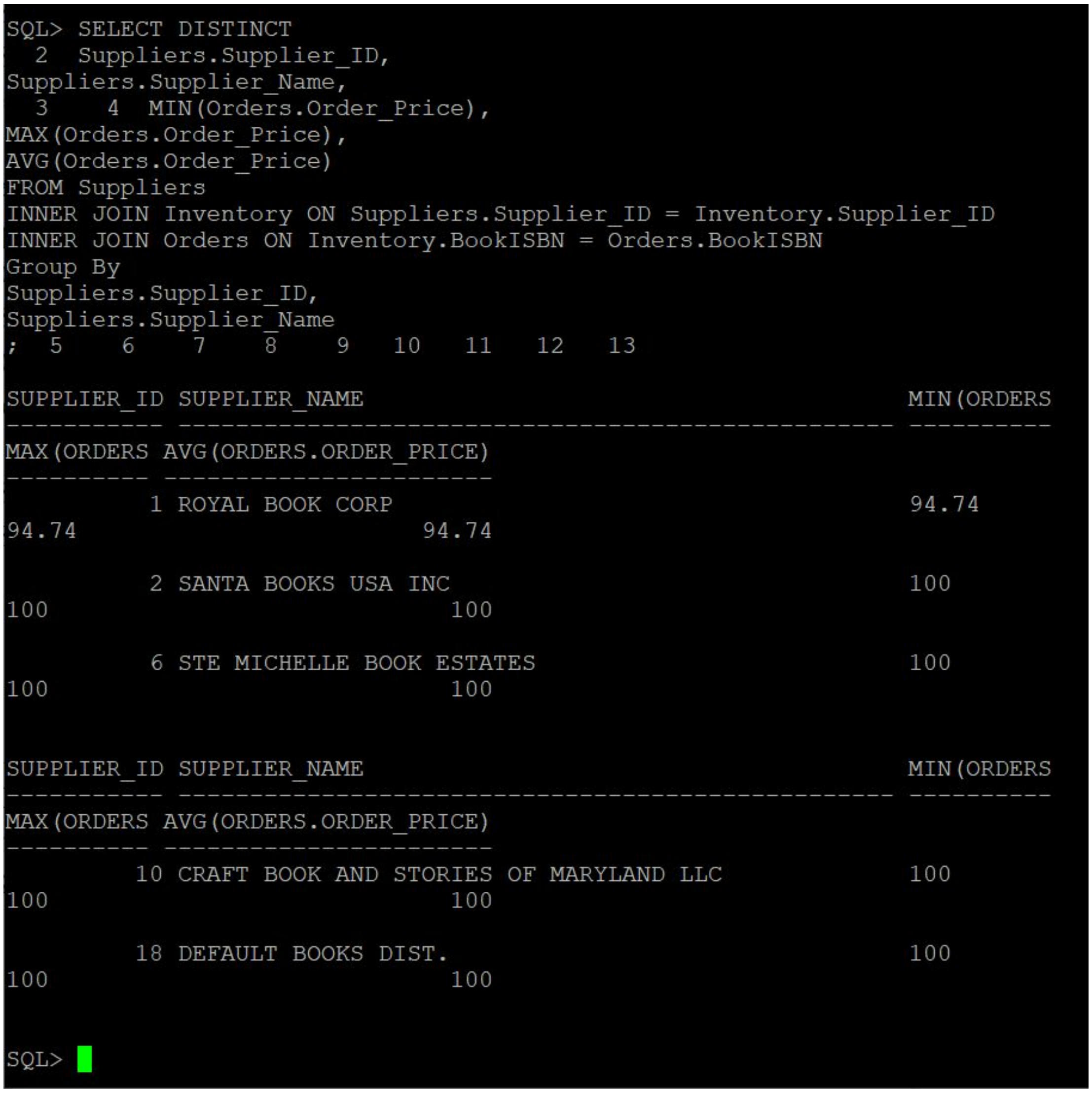
```
SELECT
Books.BookTitle,
Orders.Order_Date,
Orders.Order_Status
FROM Customers, Books, Orders
WHERE Orders.Customer_ID = Customers.Customer_ID
AND Orders.BookISBN = Books.BookISBN
AND Customers.customerfname = 'Kwai'
AND Customers.customerLname = 'Yu';
```



```
SQL>
SQL> SELECT
Books.BookTitle,
Orders.Order_Date,
Orders.Order_Status
FROM Customers, Books, Orders
WHERE Orders.Customer_ID = Customers.Customer_ID
AND Orders.BookISBN = Books.BookISBN
AND Customers.customerfname = 'Kwai'
AND Customers.customerLname = 'Yu'
;  2    3    4    5    6    7    8    9   10

BOOKTITLE
------------------------------------------------------
ORDER_DAT ORDER_STATUS
--------- ---------------
The Great Escape
24-FEB-03 Shipped


SQL>
```

ii. **Query 2: The minimum, maximum, and average order price from all suppliers.**

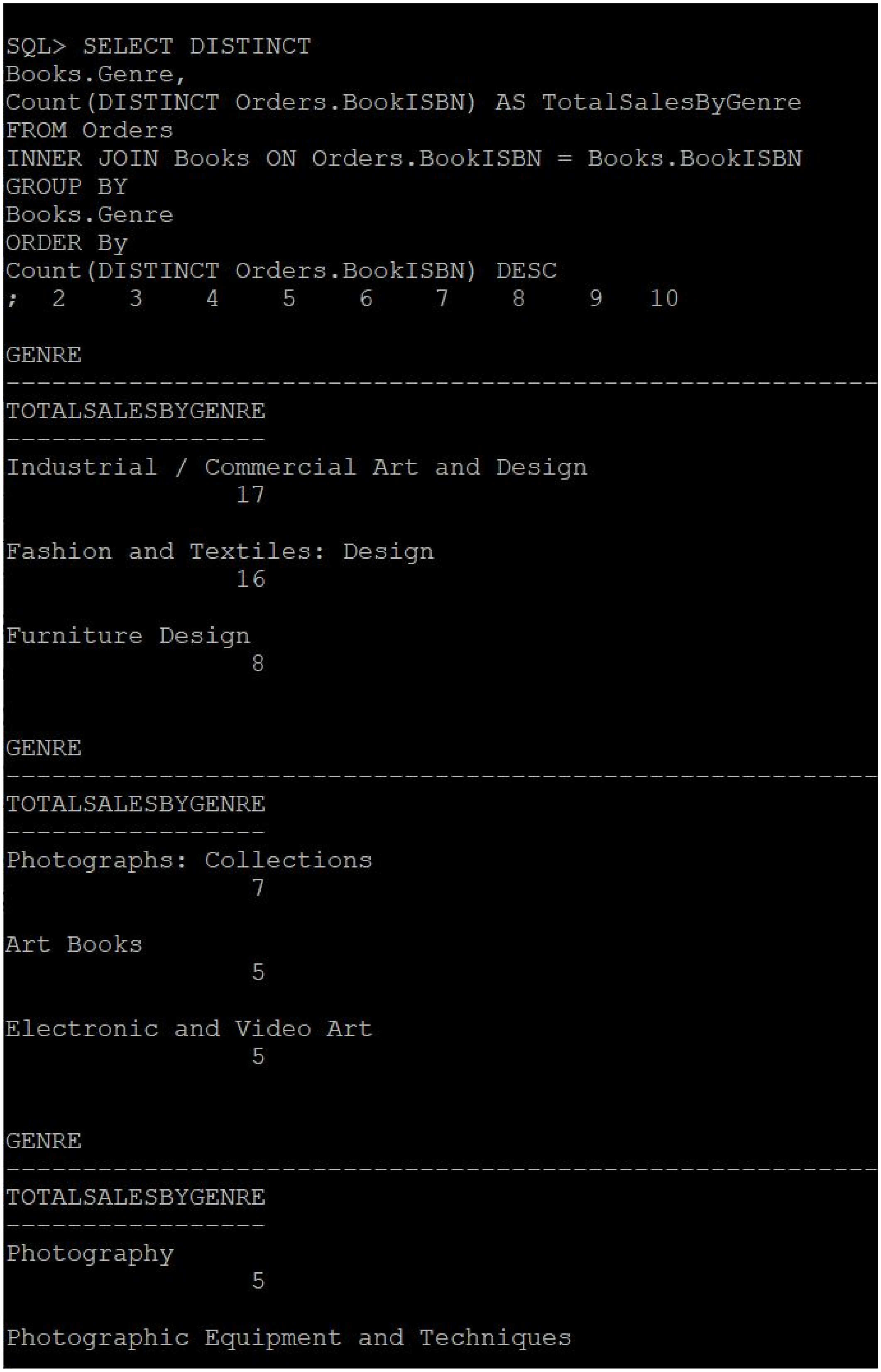
```
SELECT DISTINCT
Suppliers.Supplier_ID,
Suppliers.Supplier_Name,
MIN(Orders.Order_Price),
MAX(Orders.Order_Price),
AVG(Orders.Order_Price)
FROM Suppliers
INNER JOIN Inventory ON Suppliers.Supplier_ID =
Inventory.Supplier_ID
INNER JOIN Orders ON Inventory.BookISBN =
Orders.BookISBN
Group By
Suppliers.Supplier_ID,
Suppliers.Supplier_Name
;
```



```
SQL> SELECT DISTINCT
  2  Suppliers.Supplier_ID,
Suppliers.Supplier_Name,
  3    4  MIN(Orders.Order_Price),
MAX(Orders.Order_Price),
AVG(Orders.Order_Price)
FROM Suppliers
INNER JOIN Inventory ON Suppliers.Supplier_ID = Inventory.Supplier_ID
INNER JOIN Orders ON Inventory.BookISBN = Orders.BookISBN
Group By
Suppliers.Supplier_ID,
Suppliers.Supplier_Name
;  5    6    7    8    9   10   11   12   13

SUPPLIER_ID SUPPLIER_NAME                                      MIN(ORDERS
----------- -------------------------------------------------- ----------
MAX(ORDERS AVG(ORDERS.ORDER_PRICE)
---------- ----------------------
          1 ROYAL BOOK CORP                                    94.74
94.74                       94.74

          2 SANTA BOOKS USA INC                                100
100                           100

          6 STE MICHELLE BOOK ESTATES                          100
100                           100


SUPPLIER_ID SUPPLIER_NAME                                      MIN(ORDERS
----------- -------------------------------------------------- ----------
MAX(ORDERS AVG(ORDERS.ORDER_PRICE)
---------- ----------------------
         10 CRAFT BOOK AND STORIES OF MARYLAND LLC             100
100                           100

         18 DEFAULT BOOKS DIST.                                100
100                           100


SQL> █
```

iii. Query 3: Best selling book genres.

```
SELECT DISTINCT
Books.Genre,
Count(DISTINCT Orders.BookISBN) AS TotalSalesByGenre
FROM Orders
INNER JOIN Books ON Orders.BookISBN = Books.BookISBN
GROUP BY
Books.Genre
ORDER By
Count(DISTINCT Orders.BookISBN) DESC
;
```

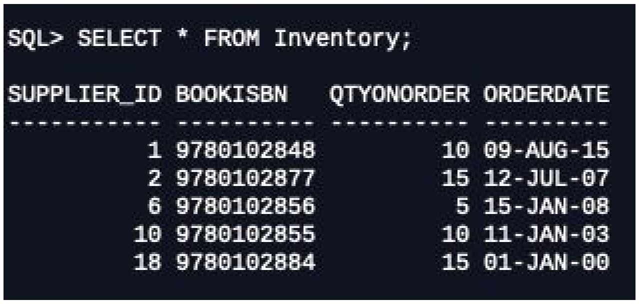


```
SQL> SELECT DISTINCT
Books.Genre,
Count(DISTINCT Orders.BookISBN) AS TotalSalesByGenre
FROM Orders
INNER JOIN Books ON Orders.BookISBN = Books.BookISBN
GROUP BY
Books.Genre
ORDER By
Count(DISTINCT Orders.BookISBN) DESC
;  2    3    4    5    6    7    8    9   10

GENRE
---------------------------------------------------------
TOTALSALESBYGENRE
-----------------
Industrial / Commercial Art and Design
               17

Fashion and Textiles: Design
               16

Furniture Design
                8


GENRE
---------------------------------------------------------
TOTALSALESBYGENRE
-----------------
Photographs: Collections
                7

Art Books
                5

Electronic and Video Art
                5


GENRE
---------------------------------------------------------
TOTALSALESBYGENRE
-----------------
Photography
                5

Photographic Equipment and Techniques
```

9. Data Manipulation:

a. DML by Ansh Chandnani

i. Table before the UPDATE command

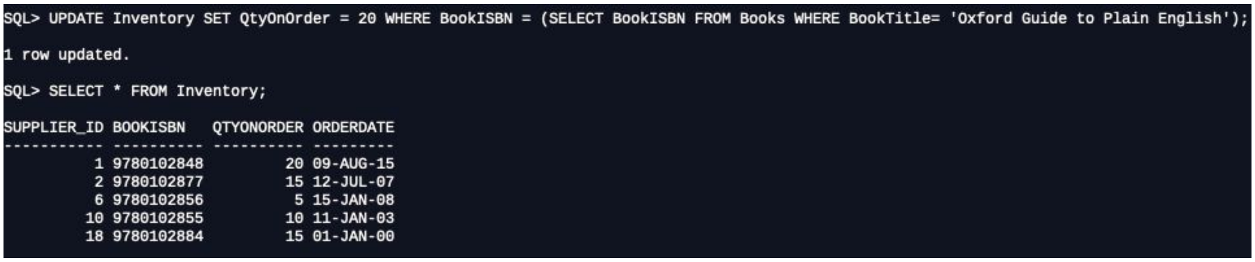


```
SQL> SELECT * FROM Inventory;

SUPPLIER_ID BOOKISBN   QTYONORDER ORDERDATE
----------- ---------- ---------- ---------
          1 9780102848         10 09-AUG-15
          2 9780102877         15 12-JUL-07
          6 9780102856          5 15-JAN-08
         10 9780102855         10 11-JAN-03
         18 9780102884         15 01-JAN-00
```

ii. UPDATE command

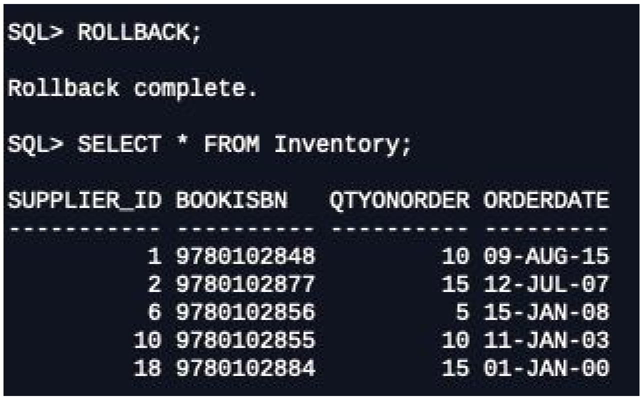
UPDATE Inventory SET QtyOnOrder = 20 WHERE BookISBN = (SELECT BookISBN FROM Books WHERE BookTitle= ‘Oxford Guide to Plain English’);

iii. Table after the UPDATE command



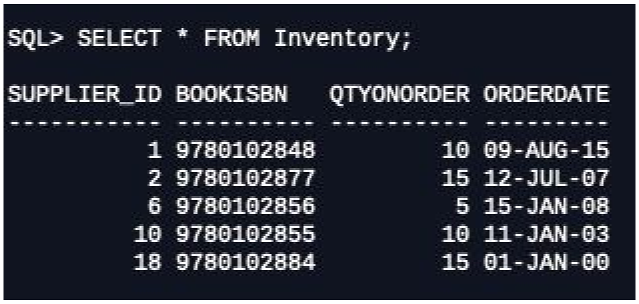
```
SQL> UPDATE Inventory SET QtyOnOrder = 20 WHERE BookISBN = (SELECT BookISBN FROM Books WHERE BookTitle= 'Oxford Guide to Plain English');

1 row updated.

SQL> SELECT * FROM Inventory;

SUPPLIER_ID BOOKISBN   QTYONORDER ORDERDATE
----------- ---------- ---------- ---------
          1 9780102848         20 09-AUG-15
          2 9780102877         15 12-JUL-07
          6 9780102856          5 15-JAN-08
         10 9780102855         10 11-JAN-03
         18 9780102884         15 01-JAN-00
```

iv. ROLLBACK



```
SQL> ROLLBACK;

Rollback complete.

SQL> SELECT * FROM Inventory;

SUPPLIER_ID BOOKISBN   QTYONORDER ORDERDATE
----------- ---------- ---------- ---------
          1 9780102848         10 09-AUG-15
          2 9780102877         15 12-JUL-07
          6 9780102856          5 15-JAN-08
         10 9780102855         10 11-JAN-03
         18 9780102884         15 01-JAN-00
```

v. **Table before the DELETE command**

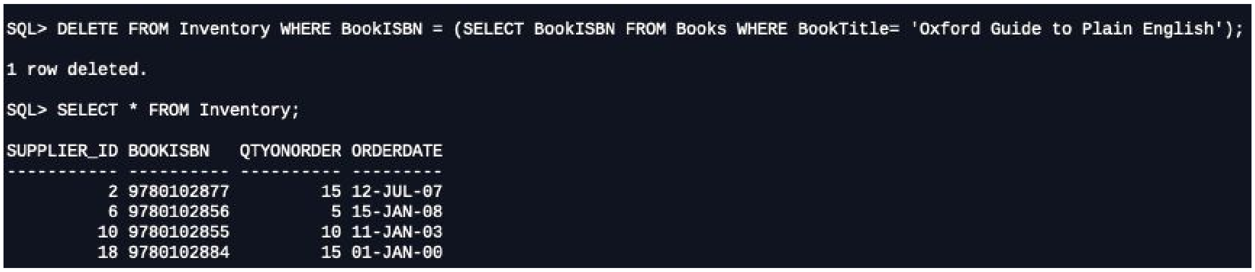


```
SQL> SELECT * FROM Inventory;

SUPPLIER_ID BOOKISBN   QTYONORDER ORDERDATE
----------- ---------- ---------- ---------
          1 9780102848         10 09-AUG-15
          2 9780102877         15 12-JUL-07
          6 9780102856          5 15-JAN-08
         10 9780102855         10 11-JAN-03
         18 9780102884         15 01-JAN-00
```

vi. **DELETE command**

```
DELETE FROM Inventory
WHERE BookISBN = (
        SELECT BookISBN
        FROM Books
        WHERE BookTitle= ‘Oxford Guide to Plain English’
);
```

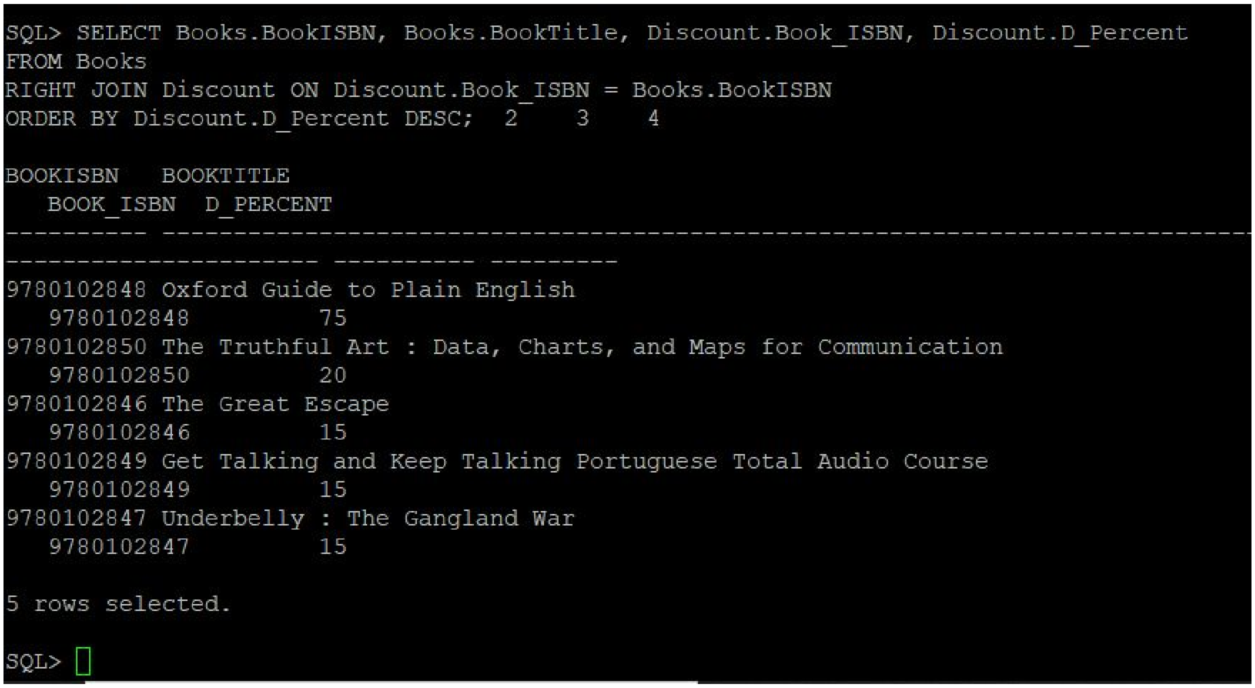
vii. **Table after the DELETE command**



```
SQL> DELETE FROM Inventory WHERE BookISBN = (SELECT BookISBN FROM Books WHERE BookTitle= 'Oxford Guide to Plain English');

1 row deleted.

SQL> SELECT * FROM Inventory;

SUPPLIER_ID BOOKISBN   QTYONORDER ORDERDATE
----------- ---------- ---------- ---------
          2 9780102877         15 12-JUL-07
          6 9780102856          5 15-JAN-08
         10 9780102855         10 11-JAN-03
         18 9780102884         15 01-JAN-00
```

b. **DML by Jordan Parton**

i. **Table before the UPDATE command**

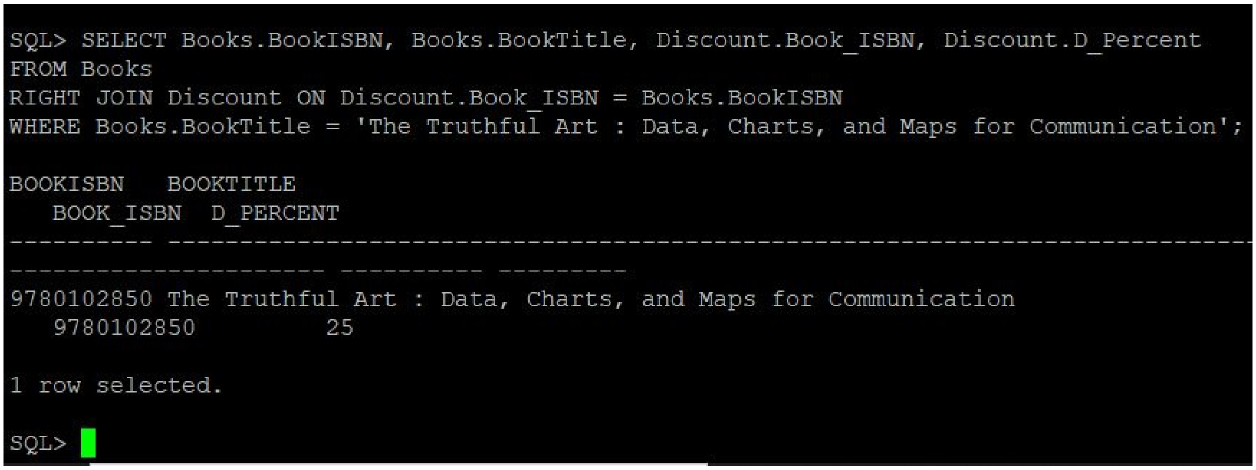


```
SQL> SELECT Books.BookISBN, Books.BookTitle, Discount.Book_ISBN, Discount.D_Percent
FROM Books
RIGHT JOIN Discount ON Discount.Book_ISBN = Books.BookISBN
ORDER BY Discount.D_Percent DESC;  2    3    4

BOOKISBN   BOOKTITLE
   BOOK_ISBN  D_PERCENT
---------- ----------------------------------------------------------------------------
---------------------- ---------- ---------
9780102848 Oxford Guide to Plain English
   9780102848         75
9780102850 The Truthful Art : Data, Charts, and Maps for Communication
   9780102850         20
9780102846 The Great Escape
   9780102846         15
9780102849 Get Talking and Keep Talking Portuguese Total Audio Course
   9780102849         15
9780102847 Underbelly : The Gangland War
   9780102847         15

5 rows selected.

SQL> 
```

ii. UPDATE command

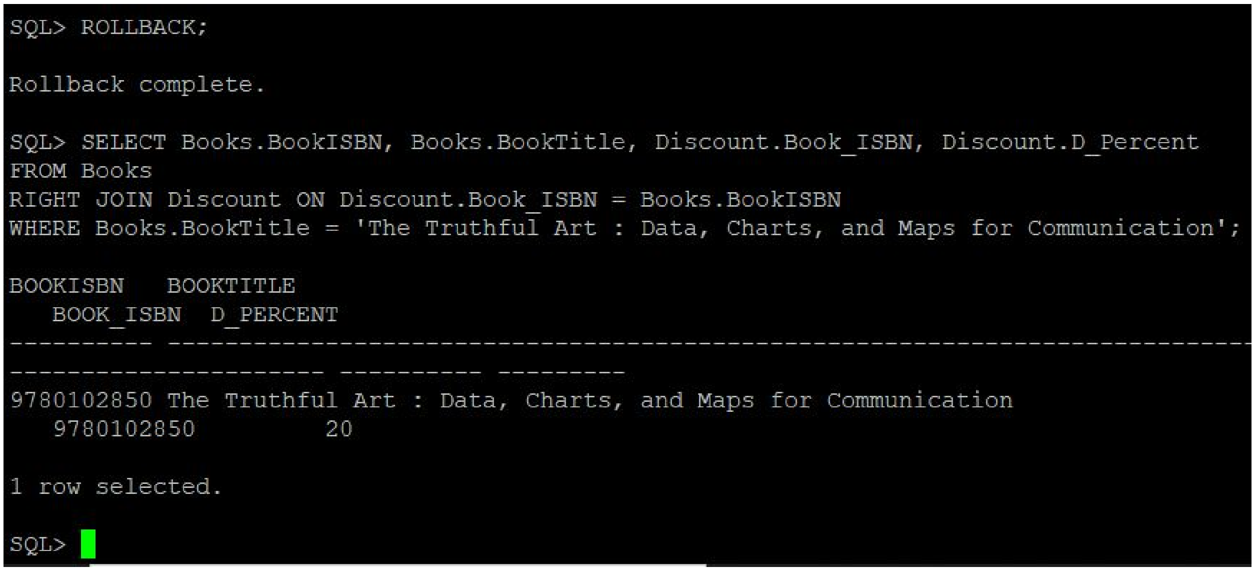
UPDATE Discount
SET D_Percent = 25
WHERE Book_ISBN = (
SELECT BookISBN FROM Books
WHERE BookTitle = 'The Truthful Art : Data, Charts, and Maps for Communication');

iii. Table after the UPDATE command



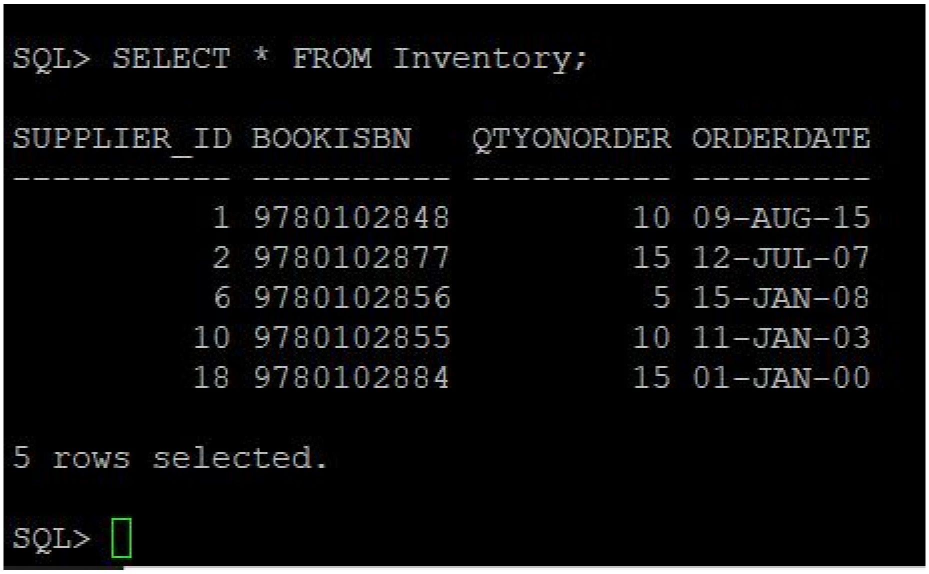
```
SQL> SELECT Books.BookISBN, Books.BookTitle, Discount.Book_ISBN, Discount.D_Percent
FROM Books
RIGHT JOIN Discount ON Discount.Book_ISBN = Books.BookISBN
WHERE Books.BookTitle = 'The Truthful Art : Data, Charts, and Maps for Communication';

BOOKISBN   BOOKTITLE
   BOOK_ISBN  D_PERCENT
---------- ----------------------------------------------------------------------------
---------------------- ---------- ---------
9780102850 The Truthful Art : Data, Charts, and Maps for Communication
   9780102850         25

1 row selected.

SQL> 
```

iv. ROLLBACK



```
SQL> ROLLBACK;

Rollback complete.

SQL> SELECT Books.BookISBN, Books.BookTitle, Discount.Book_ISBN, Discount.D_Percent
FROM Books
RIGHT JOIN Discount ON Discount.Book_ISBN = Books.BookISBN
WHERE Books.BookTitle = 'The Truthful Art : Data, Charts, and Maps for Communication';

BOOKISBN   BOOKTITLE
   BOOK_ISBN  D_PERCENT
---------- --------------------------------------------------------------------------------
---------------------- ---------- ---------
9780102850 The Truthful Art : Data, Charts, and Maps for Communication
   9780102850         20

1 row selected.

SQL> 
```

v. **Table before the DELETE command**

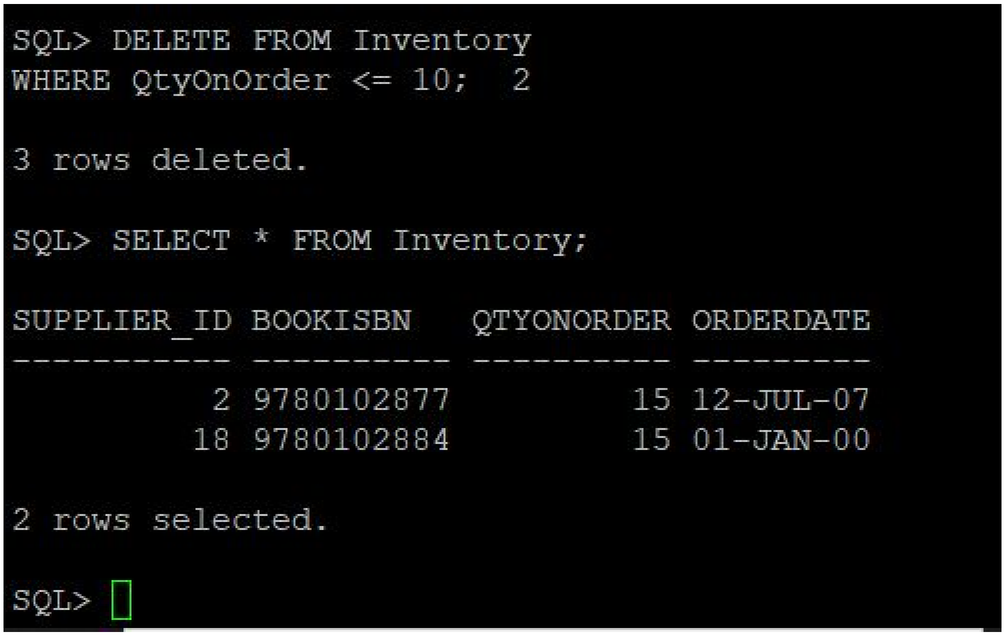


```
SQL> SELECT * FROM Inventory;

SUPPLIER_ID BOOKISBN   QTYONORDER ORDERDATE
----------- ---------- ---------- ---------
          1 9780102848         10 09-AUG-15
          2 9780102877         15 12-JUL-07
          6 9780102856          5 15-JAN-08
         10 9780102855         10 11-JAN-03
         18 9780102884         15 01-JAN-00

5 rows selected.

SQL> 
```

vi. **DELETE command**

```
DELETE FROM Inventory
WHERE QtyOnOrder <= 10;
```

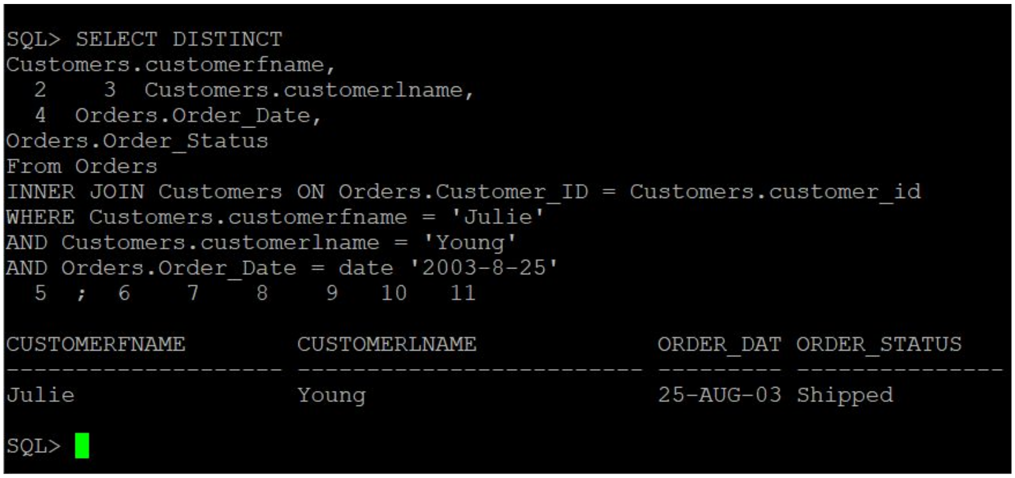
vii. **Table after the DELETE command**



```
SQL> DELETE FROM Inventory
WHERE QtyOnOrder <= 10;  2

3 rows deleted.

SQL> SELECT * FROM Inventory;

SUPPLIER_ID BOOKISBN   QTYONORDER ORDERDATE
----------- ---------- ---------- ---------
          2 9780102877         15 12-JUL-07
         18 9780102884         15 01-JAN-00

2 rows selected.

SQL> 
```

c. **DML By Sara-Jo Fegley:**

i. **Table before the UPDATE command**



```
SQL> SELECT DISTINCT
Customers.customerfname,
  2    3  Customers.customerlname,
  4  Orders.Order_Date,
Orders.Order_Status
From Orders
INNER JOIN Customers ON Orders.Customer_ID = Customers.customer_id
WHERE Customers.customerfname = 'Julie'
AND Customers.customerlname = 'Young'
AND Orders.Order_Date = date '2003-8-25'
  5  ;  6    7    8    9   10   11

CUSTOMERFNAME        CUSTOMERLNAME             ORDER_DAT ORDER_STATUS
-------------------- ------------------------- --------- ---------------
Julie                Young                     25-AUG-03 Shipped

SQL> 
```

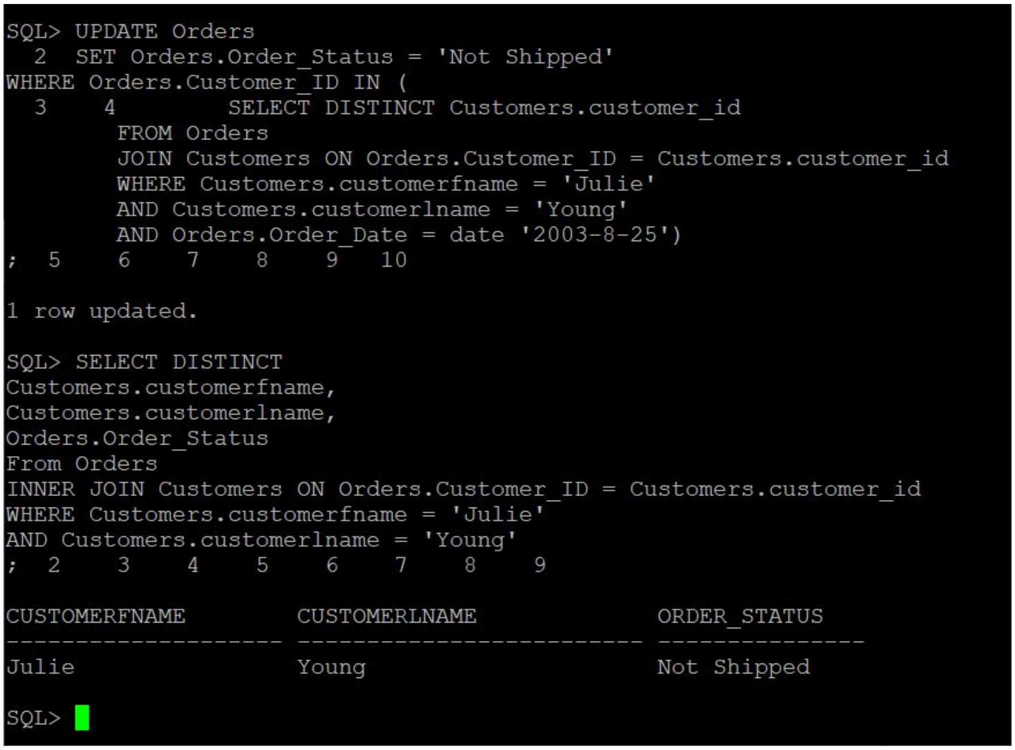
ii. **UPDATE command: Update Julie Young's order on August 25, 2003 as 'Not Shipped'.**

```
UPDATE Orders
SET Orders.Order_Status = 'Not Shipped'
WHERE Orders.Customer_ID IN (
        SELECT DISTINCT Customers.customer_id
        FROM Orders
        JOIN Customers ON Orders.Customer_ID =
Customers.customer_id
        WHERE Customers.customerfname = 'Julie'
```

AND Customers.customerlname = 'Young'
AND Orders.Order_Date = date '2003-8-25')

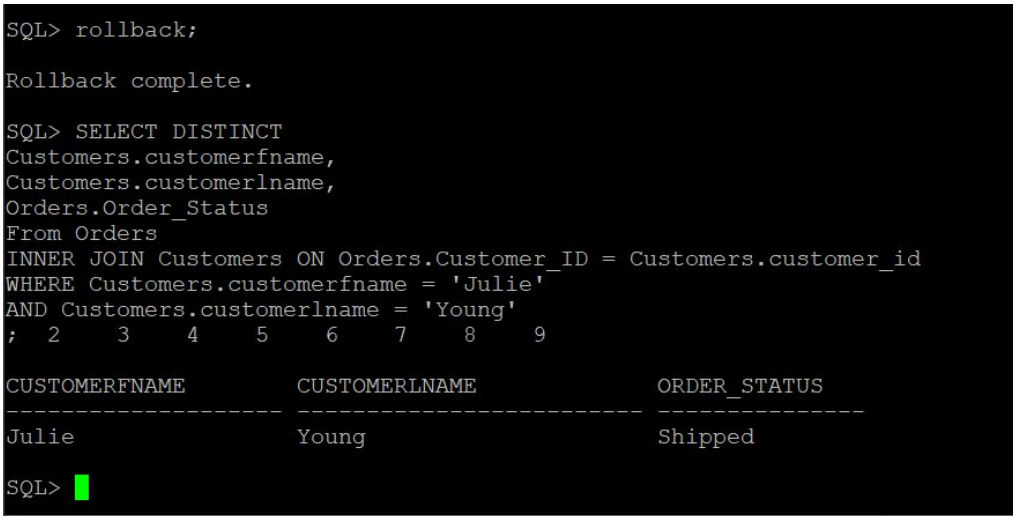
;

iii. Table after the UPDATE command



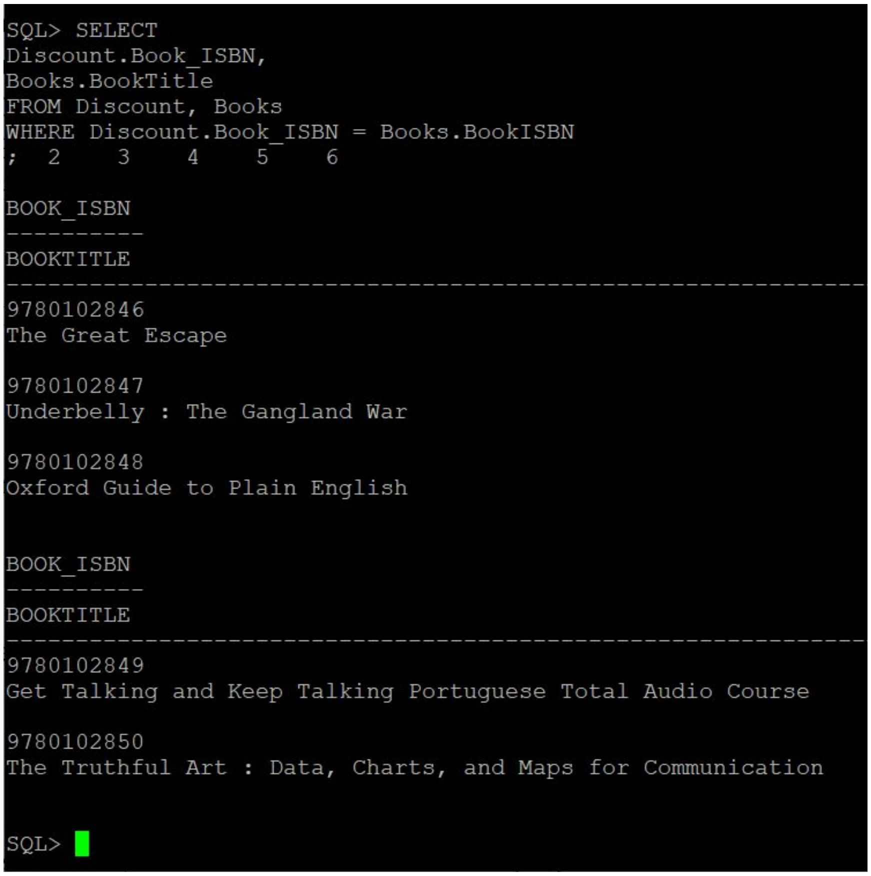
```
SQL> UPDATE Orders
  2  SET Orders.Order_Status = 'Not Shipped'
WHERE Orders.Customer_ID IN (
  3    4        SELECT DISTINCT Customers.customer_id
        FROM Orders
        JOIN Customers ON Orders.Customer_ID = Customers.customer_id
        WHERE Customers.customerfname = 'Julie'
        AND Customers.customerlname = 'Young'
        AND Orders.Order_Date = date '2003-8-25')
;  5    6    7    8    9   10

1 row updated.

SQL> SELECT DISTINCT
Customers.customerfname,
Customers.customerlname,
Orders.Order_Status
From Orders
INNER JOIN Customers ON Orders.Customer_ID = Customers.customer_id
WHERE Customers.customerfname = 'Julie'
AND Customers.customerlname = 'Young'
;  2    3    4    5    6    7    8    9

CUSTOMERFNAME        CUSTOMERLNAME             ORDER_STATUS
-------------------- ------------------------- ---------------
Julie                Young                     Not Shipped

SQL>
```

iv. ROLLBACK



```
SQL> rollback;

Rollback complete.

SQL> SELECT DISTINCT
Customers.customerfname,
Customers.customerlname,
Orders.Order_Status
From Orders
INNER JOIN Customers ON Orders.Customer_ID = Customers.customer_id
WHERE Customers.customerfname = 'Julie'
AND Customers.customerlname = 'Young'
;  2    3    4    5    6    7    8    9

CUSTOMERFNAME        CUSTOMERLNAME             ORDER_STATUS
-------------------- ------------------------- ---------------
Julie                Young                     Shipped

SQL>
```

v. Table before the DELETE command.

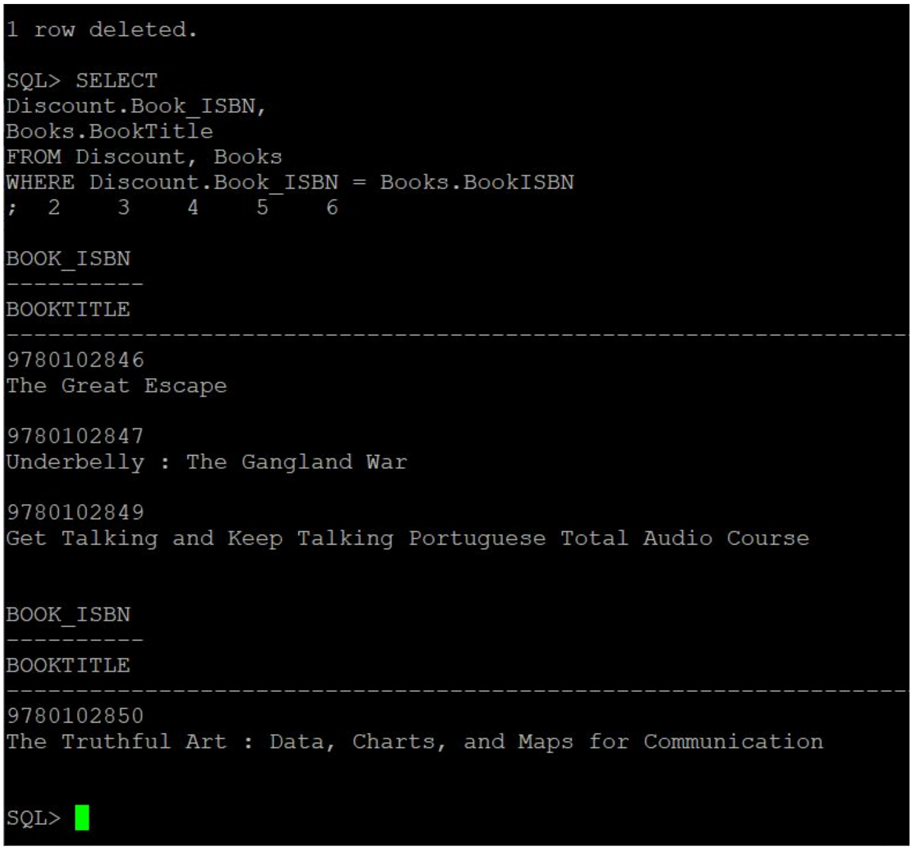


```
SQL> SELECT
Discount.Book_ISBN,
Books.BookTitle
FROM Discount, Books
WHERE Discount.Book_ISBN = Books.BookISBN
;  2    3    4    5    6

BOOK_ISBN
----------
BOOKTITLE
--------------------------------------------------------------
9780102846
The Great Escape

9780102847
Underbelly : The Gangland War

9780102848
Oxford Guide to Plain English


BOOK_ISBN
----------
BOOKTITLE
--------------------------------------------------------------
9780102849
Get Talking and Keep Talking Portuguese Total Audio Course

9780102850
The Truthful Art : Data, Charts, and Maps for Communication


SQL> 
```

vi. **DELETE command: Delete discount for book title ‘Oxford Guide to Plain English’ from Discount table.**

```
WHERE Discount.Book_ISBN =
        (SELECT BookISBN FROM Books
        WHERE BookTitle = 'Oxford Guide to Plain English')
        ;
```

vii. Table after the DELETE command



```
1 row deleted.

SQL> SELECT
Discount.Book_ISBN,
Books.BookTitle
FROM Discount, Books
WHERE Discount.Book_ISBN = Books.BookISBN
;  2    3    4    5    6

BOOK_ISBN
----------
BOOKTITLE
----------------------------------------------------------------
9780102846
The Great Escape

9780102847
Underbelly : The Gangland War

9780102849
Get Talking and Keep Talking Portuguese Total Audio Course


BOOK_ISBN
----------
BOOKTITLE
----------------------------------------------------------------
9780102850
The Truthful Art : Data, Charts, and Maps for Communication


SQL>
```

10. Summary:

a. **Summary by Ansh Chandnani**

Having some prior experience with SQL, this project was the next iterative step in my Database and SQL Skills. I learned about multiple theoretical concepts and their significance in relational databases such as integrity. This project was the first time I learned how to create Entity-Relationship Diagrams and decompose relationships like many-to-many relationships and organize databases according to the industry standards. I now feel confident working with relational databases and understand the advantages and disadvantages of relational database models and their use in the tech industry.

b. **Summary by Jordan Parton**

Over the course of the project, I was able to apply the skills I learned in class, as well as learn from my team on how to actually utilize SQL to its fullest extent. Having this class as an entry into SQL and databases in general will help me in the future as now, I feel as though I have a much better understanding of the language and what building and maintaining a database actually means. Not only does it involve getting data and keeping it up to date, it also involves much more planning and thinking about how the data will be used, and how an end user may query the database. I now feel confident that I can successfully work on building and maintaining a database outside of the classroom.

c. **Summary by Sara-Jo Fegley**

This project has enforced the multiple SQL and database skills that I have learned throughout this term. The ERD was extremely helpful in not only the deployment of the tables but also statements. Therefore, I have realized the importance in the complete development of a relational database's logical structure for future projects. If granted more time with this project, I would have attempted to create different views for common user queries.

11. Appendix:

a. **Proposed Deliverables and Division of Work:**
We will develop all of the necessary documentation outlined in the project handout. The documentation will discuss our experience with the project itself, as well as everything we have learned about constructing and managing a database. Using each of our own knowledge, we will agree on an ERD with all attributes in each table, and then will each construct one of the proposed tables. Each member is tasked with the maintenance of a single table within the database. We will then work as a team to join the tables together into a complete and functional database which will be submitted to Blackboard.

b. **ERD Contributions:**
   - i. Ansh Chandnani: I created the first ERD that included all tables. After which I created the 'Order', 'Customers', and 'Discount' tables
   - ii. Sara-Jo Fegley: I created the 'Books', 'Book_Author', and 'Author' table
   - iii. Jordan Parton: I created and maintained the 'Suppliers' and 'Inventory' tables

12. Known References

a. Allen, Christopher, et al. *Introduction to Relational Databases and SQL Programming*. McGraw Hill Technology Education, 2004.

b. Letts, M. (2000). iUniverse provides outsourced services to IDG Books - Trade publisher to use iUniverse to manage its library database and online bookstore. *Seybold Report on Internet Publishing.*, *4*(11).

c. Mkondratek. "Mkondratek/Bookstore-Database-Design." *GitHub*, 3 Oct. 2018, github.com/mkondratek/Bookstore-Database-Design.

d. Rules and Policies - Protecting PII - Privacy Act. (2018, September 24). Retrieved from https://www.gsa.gov/reference/gsa-privacy-program/rules-and-policies-protecting-pii-privacy-act