Assignment 10.1: Database Documentation

Jasmine K Sanders

Sullivan University

CSC210: Database Design

Prof. Mead

December 14, 2025

Assignment 10.1: Database Documentation

This document outlines the design and creation process for the Online Apparel Store Consumer Data database. I created this database with the intention of implementing it at my current workplace when we transition the online apparel store from third-party operated to in-house. This database is meant to capture consumer data from the online store, so we may analyze the information to determine how to update or improve the online apparel store to maximize profits. This information will help the store to determine if/when to alter prices, items offered, shipping carriers, or other components related to the online store. It will also help with tracking buying habits of consumers, so the store may capitalize on current consumer trends.

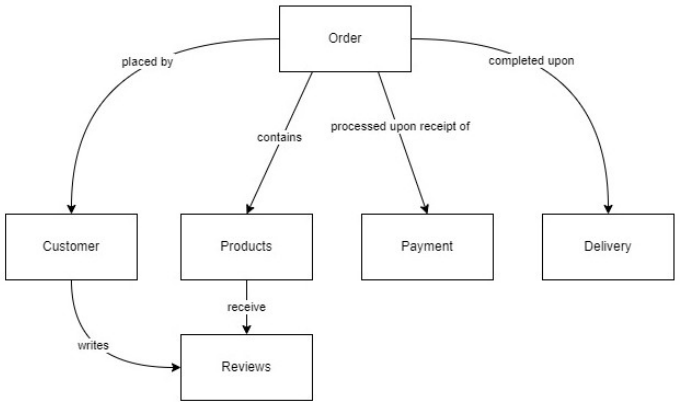
Building the Data Models

Conceptual Data Model

A conceptual data model provides a high-level view of entities and their relationships without technical details (Rivera, 2025). To create the Conceptual Data Model development process begins with identifying the entities and their relationships. The entities for this database include (1) Orders, (2) Customers, (3) Products, (4) Payment, (5) Delivery, and (6) Reviews. The general relationships between each of these tables are as follows: each order is placed by a customer, each order contains products, each order is processed upon receipt of a payment, each order is considered complete upon delivery, each customer may write a review, and each product may receive reviews. These entities and relationships are displayed below, in figure 1.

Figure 1

Conceptual Data Model for an Online Apparel Store's Consumer Data Database



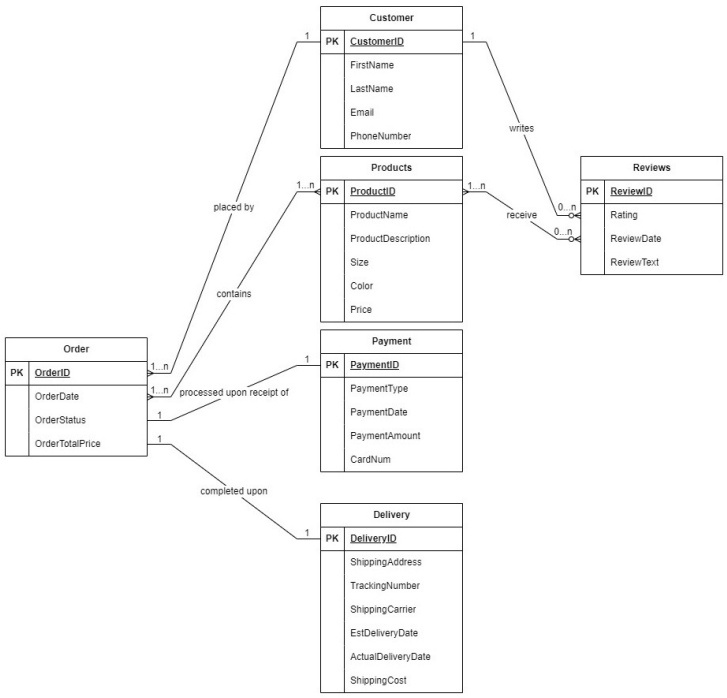
Logical Data Model

A logical data model is more detailed than a conceptual data model. It builds on the conceptual model by adding specific attributes and defining relationships more precisely, but it remains independent of any specific database technology (Rivera, 2025). The Logical Data Model development process begins with identifying the key entities. For this database, the entities are Order, Customer, Products, Payment, Delivery, and Reviews. The next step includes identifying the relationships and types of relationships between these entities. This phase focuses on defining what data is important without considering how it will be implemented technically. For this database the relationships between each of the tables are as follows: each order is placed by a customer (one-to-one), each order contains one or more products and each product may be in one or more orders (many-to-many), each order is processed upon receipt of a payment (one-to-one), each order is considered complete upon delivery (one-to-one), each customer may write zero, one, or may reviews (one-to-many optional), and each product may receive zero, one, or many reviews but each review could refer to one or many products (many-to-many optional). These relationships establish the structure of the model. Following this, attributes are assigned to each entity to capture essential details for identification and categorization. Attributes can be further refined by specifying primary keys (unique identifiers). Attributes are listed within each entity's table in figure 2. For example, the Customer's attributes are CustomerID, FirstName,

LastName, Email, and Phone Number. The CustomerID attribute is the primary key for the Customer entity. Please refer to figure 2 for additional attributes and primary keys.

Figure 2

Logical Data Model for an Online Apparel Store's Consumer Data Database



Physical Data Model

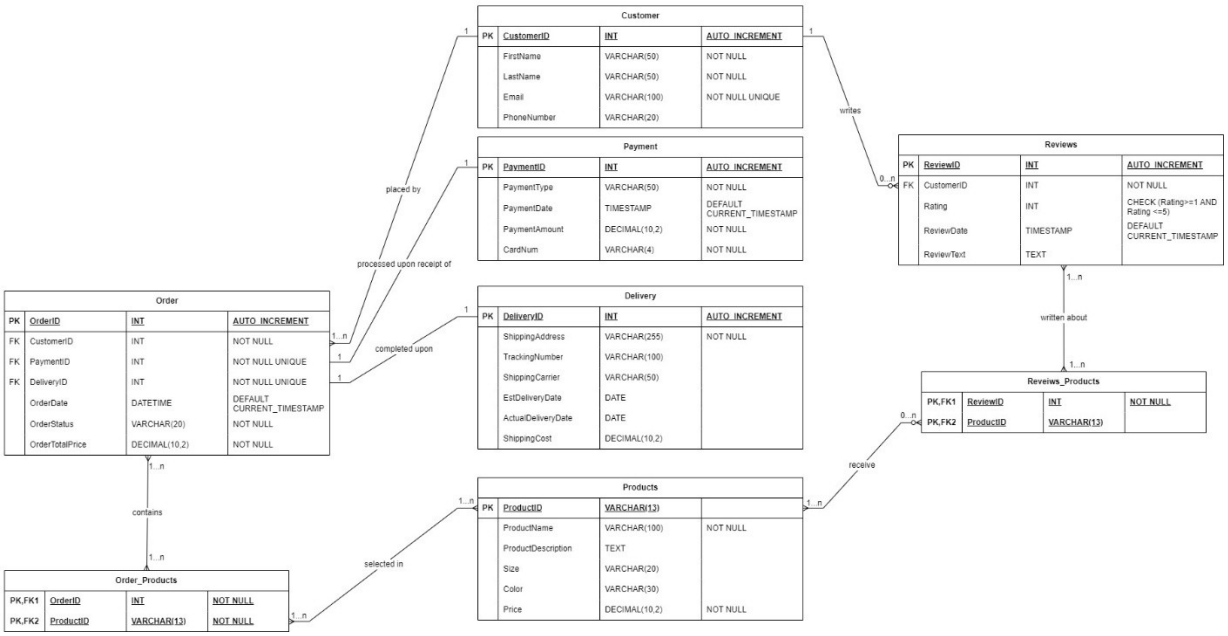
The physical data model builds upon the conceptual and logical data models by detailing how the data will be stored within a database, outlining the technical details needed to create and manage a database (Rivera, 2025). In figure 3, not only are primary keys and table relationships identified, but foreign keys, data types, constraints, and junction tables are identified as well.

While creating this model, I realized that I could not record the entire card number in the database, because it is illegal to store debit/credit card information in an unsecure database. Because of this, I opted to only record the last 4 digits of a credit/debit card number, which is

common practice due to the legal limitations. Therefore, the card number (CardNum) data input to 4 characters.

Figure 3

Physical Data Model for an Online Apparel Store's Consumer Data Database



Tools Used for Data Modeling

Data modeling relies on tools and techniques that facilitate the visualization, organization, and refinement of data structures. To create the Conceptual Data Model, Logical Data Model, and Physical Data Model previously mentioned, Draw.io was instrumental.

For the Conceptual Data Model, the Line, Rectangle, Text, and Title tools were used to create the model - beginning with the Rectangles and Title boxes to form the entities then finalizing the relationships with the line and text tools.

For the Logical Data Model, the Table 1, 1 to 1, 1 to Many, and text tools included under the Entity Relation dropdown selection were used and formatted to create the model. First, the Table 1 Entity Relation table was used to identify the entities. Then, the 1 to 1 and 1 to Many

line tools were used to establish the relationships between each entity. Some of these relationships had to be manually formatted to accommodate the different relationship types. Then the text tool was used to label the relationship lines to showcase the type of relationship (1 to 1, 1 to many, etc), and to label how each object relates. Next, the Table 1 tools were edited to include the respective attributes and identify the PKs (primary keys).

The Physical Data Model was built using similar methods to the Logical Model, except the Table 1 tool was edited to include additional columns for the data types and other data constraints. Plus, the Table 2 tool was used and edited to display the junction tables.

Building the Database from the Data Models

To begin creating and refining the database, I first created a database with tables and attributes based on the Logical and Physical Data Models for the Online Apparel Store's Consumer Data Database.

To create the database with tables and attributes, I utilized the DDL queries CREATE DATABASE and CREATE TABLE as shown in figures 4, 5 and 6. During this process, each attribute was assigned a datatype in accordance with the type of data that would be associated with each attribute. I also assigned specific attributes as PRIMARY KEYS based on the attributes chosen as primary keys in figure 2: the Logical Data Model for the Online Apparel Store's Consumer Data Database. Furthermore, I added the constraint NOT NULL to certain attributes to signify that the values for those attributes are mandatory and cannot be empty/null. I also used the constraint UNIQUE to ensure all values in specific columns are different. For OrderDate and PaymentDate, I added the constraint DEFAULT CURRENT_TIMESTAMP to set the current date and time as the default value for each column. Furthermore, for the Rating attribute, I added the constraint CHECK (Rating>=1 AND Rating<=5) to ensure the rating

values entered are within the 1-5 range, because rating values outside of the 1-5 range should not be possible for the rating method utilized in the online apparel store.

Next, I established many-to-many relationships through junction tables and foreign keys for the orders and products relationship as well as the reviews and products relationship. To accomplish this, I utilized CREATE TABLE, PRIMARY KEY, and FOREIGN KEY as shown in figure 6. When creating the junction tables, I identified the primary keys and established foreign key constraints in an effort to enforce referential integrity.

Afterwards, I utilized the DML query INSERT INTO to insert data values into the tables. This method allowed me to insert data records by specifically providing values for each column as shown in figures 7, 8, and 9. Once the data was added, I created a view labeled “AllTables” to display the relationships for all the records from all the tables for the database. The “AllTables” view is shown in Tables 1, 2, 3, and 4. To create the “AllTables” view, I used CREATE VIEW AS, SELECT, and LEFT JOIN as shown in figures 10 and 11. With this view, one is able to ensure the relationships outlined in the data models are properly reflected in the database.

Conclusion

Altogether the Online Apparel Store Consumer Data database will capture consumer data for analyzing and optimizing store operations, including pricing, product offerings, and consumer behavior trends related to the online apparel store as it transitions from third-party management to in-house operation. The development of the database involved three data modeling stages. The conceptual model which identifies entities (Orders, Customers, Products, Payment, Delivery, Reviews) and their relationships. The logical model which details these relationships with attributes, including primary keys. The physical model which specifies how data is stored and managed. Then, the database was created and refined based on the previously

created data models, utilizing DDL queries to define tables, attributes, and constraints to ensure data integrity and enforce relationships through junction tables and foreign keys as well as DML queries to insert data values into the database.

References

Rivera, S. (2025, August 5). *Conceptual vs logical VS physical data models: What's the difference?* ThoughtSpot. https://www.thoughtspot.com/data-trends/data-modeling/conceptual-vs-logical-vs-physical-data-models

Appendix

Figure 1

Conceptual Data Model for an Online Apparel Store's Consumer Data Database

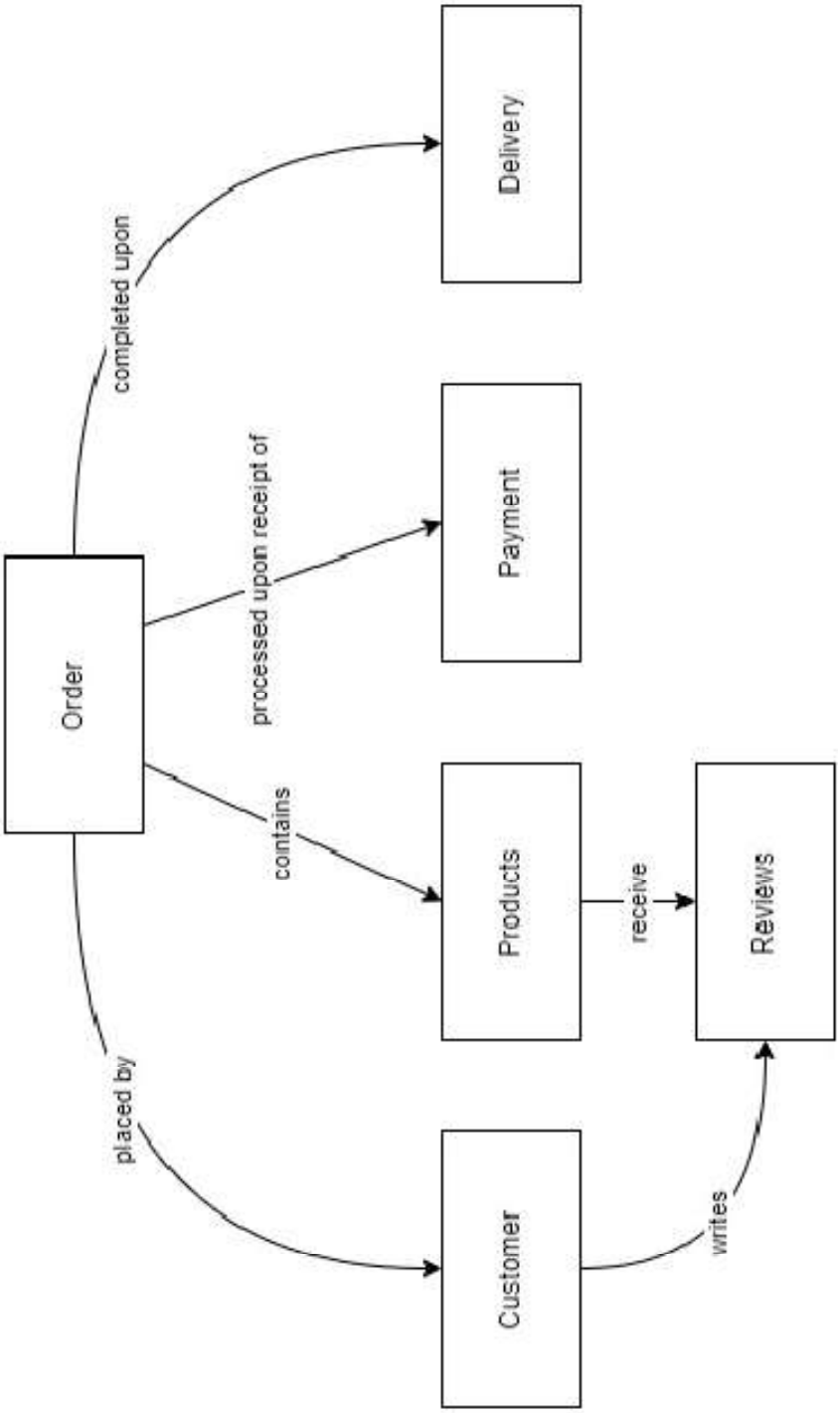


Figure 2

Logical Data Model for an Online Apparel Store's Consumer Data Database

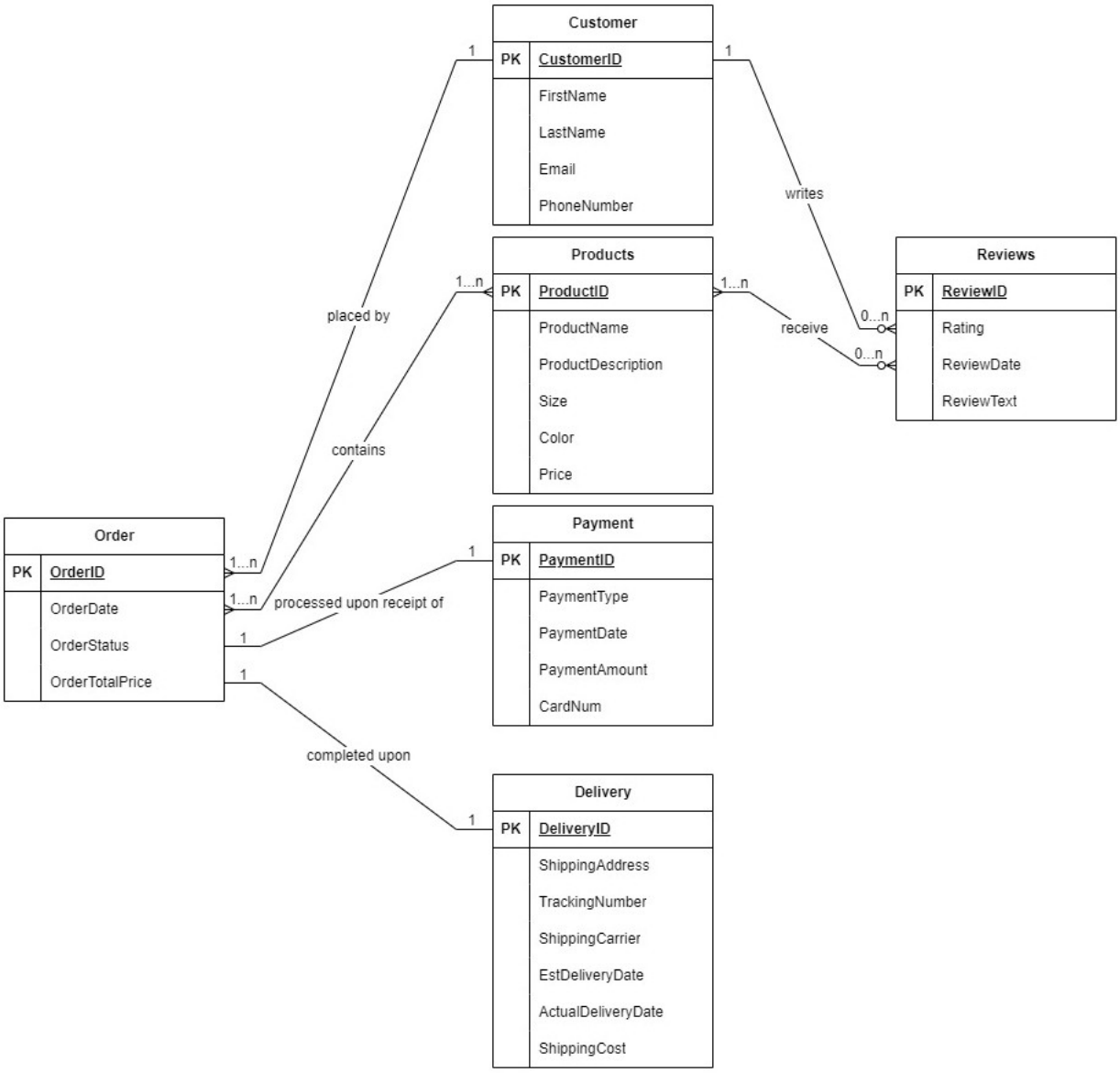


Figure 3

Physical Data Model for an Online Apparel Store's Consumer Data Database

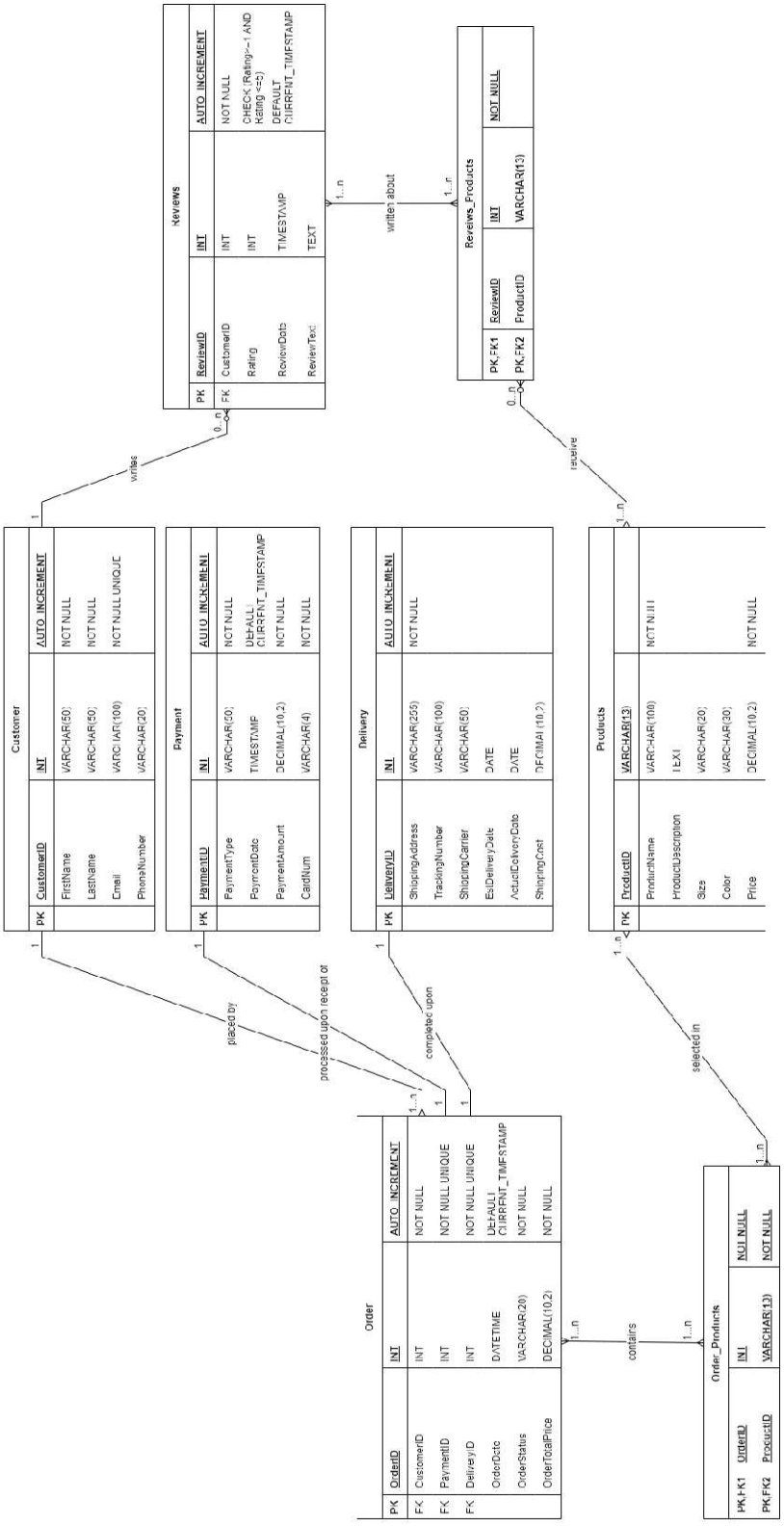


Figure 4

Online Apparel Store's Consumer Data Database Script Lines 1-29

```
1   -- Create the database
2   CREATE DATABASE OnlineApparelStoreConsumerData;
3   USE OnlineApparelStoreConsumerData;
4
5   -- Create Customer table
6   CREATE TABLE Customer (
7       CustomerID INT AUTO_INCREMENT PRIMARY KEY,
8       FirstName VARCHAR(50) NOT NULL,
9       LastName VARCHAR(50) NOT NULL,
10      Email VARCHAR(100) NOT NULL UNIQUE,
11      PhoneNumber VARCHAR(20)
12  );
13
14  -- Create Payment table
15  CREATE TABLE Payment (
16      PaymentID INT AUTO_INCREMENT PRIMARY KEY,
17      PaymentType VARCHAR(50) NOT NULL,
18      PaymentDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
19      PaymentAmount DECIMAL(10,2) NOT NULL,
20      CardNum VARCHAR(4) NOT NULL
21  );
22
23  -- Create Delivery table
24  CREATE TABLE Delivery (
25      DeliveryID INT AUTO_INCREMENT PRIMARY KEY,
26      ShippingAddress VARCHAR(255) NOT NULL,
27      TrackingNumber VARCHAR(100),
28      ShippingCenter VARCHAR(50),
29      EstDeliveryDate DATE,
```

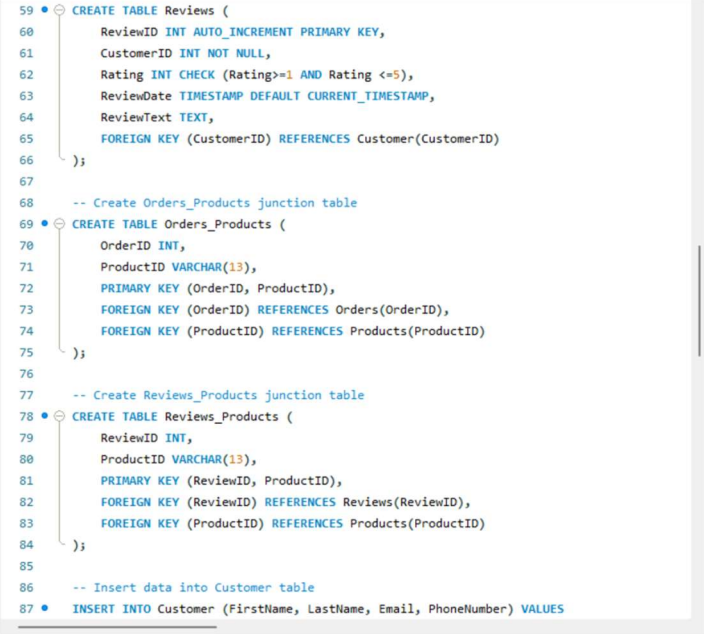
Figure 5

Online Apparel Store's Consumer Data Database Script Lines 30-58

```
30      ActualDeliveryDate DATE,
31      ShippingCost DECIMAL(10,2)
32  );
33
34  -- Create Products table
35  CREATE TABLE Products (
36      ProductID VARCHAR(13) PRIMARY KEY,
37      ProductName VARCHAR(100) NOT NULL,
38      ProductDescription TEXT,
39      Size VARCHAR(20),
40      Color VARCHAR(30),
41      Price DECIMAL(10,2) NOT NULL
42  );
43
44  -- Create Order table
45  CREATE TABLE Orders (
46      OrderID INT AUTO_INCREMENT PRIMARY KEY,
47      CustomerID INT NOT NULL,
48      PaymentID INT NOT NULL UNIQUE,
49      DeliveryID INT NOT NULL UNIQUE,
50      OrderDate DATETIME DEFAULT CURRENT_TIMESTAMP,
51      OrderStatus VARCHAR(20) NOT NULL,
52      OrderTotalPrice DECIMAL(10,2) NOT NULL,
53      FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
54      FOREIGN KEY (PaymentID) REFERENCES Payment(PaymentID),
55      FOREIGN KEY (DeliveryID) REFERENCES Delivery(DeliveryID)
56  );
57
58  -- Create Reviews table
```

Figure 6

Online Apparel Store's Consumer Data Database Script Lines 59-87



```
59  CREATE TABLE Reviews (
60      ReviewID INT AUTO_INCREMENT PRIMARY KEY,
61      CustomerID INT NOT NULL,
62      Rating INT CHECK (Rating>=1 AND Rating <=5),
63      ReviewDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
64      ReviewText TEXT,
65      FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
66  );
67
68  -- Create Orders_Products junction table
69  CREATE TABLE Orders_Products (
70      OrderID INT,
71      ProductID VARCHAR(13),
72      PRIMARY KEY (OrderID, ProductID),
73      FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
74      FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
75  );
76
77  -- Create Reviews_Products junction table
78  CREATE TABLE Reviews_Products (
79      ReviewID INT,
80      ProductID VARCHAR(13),
81      PRIMARY KEY (ReviewID, ProductID),
82      FOREIGN KEY (ReviewID) REFERENCES Reviews(ReviewID),
83      FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
84  );
85
86  -- Insert data into Customer table
87  INSERT INTO Customer (FirstName, LastName, Email, PhoneNumber) VALUES
```

Figure 7

Online Apparel Store's Consumer Data Database Script Lines 88-116

```
88      ('Alice', 'Smith', 'alice.smith@example.com', '541-555-4567'),
89      ('Bob', 'Johnson', 'bob.j@example.com', '817-555-6543'),
90      ('Alex', 'Wayne', 'AWAYNE03@example.com', '742-555-2418'),
91      ('Myles', 'Keaton', 'Keaton64@example.com', '732-555-7714'),
92      ('Laura', 'Hall', 'Laura.Hall641@example.com', '791-555-6351');
93
94      -- Insert data into Payment table
95  INSERT INTO Payment (PaymentType, PaymentDate, PaymentAmount, CardNum) VALUES
96      ('Visa', '2025-11-20 10:30:00', 75.99, '4432'),
97      ('MasterCard', '2025-11-21 14:00:00', 120.50, '6651'),
98      ('Visa', '2025-01-28 20:50:03', 17.40, '8642'),
99      ('Discover', '2025-04-07 23:10:46', 18.95, '9528'),
100     ('AmEx', '2025-10-19 16:19:40', 212.53, '6458');
101
102     -- Insert data into Delivery table
103 INSERT INTO Delivery (ShippingAddress, TrackingNumber, ShippingCenter, EstDeliveryDate, ActualDeliveryDate, ShippingCost) VALUES
104     ('123 Main St, Lebanon, NH', 'TRK123456789', 'UPS', '2025-11-23', '2025-11-24', 12.99),
105     ('456 Oak Ave, Fredericksburg, KY', 'TRK627654321', 'UPS', '2025-11-24', '2025-11-24', 32.99),
106     ('P.O. Box 685, 9296 Orci. Street, Georgetown, TN', 'TRK757638241', 'FedEx', '2025-01-30', '2025-01-29', 17.50),
107     ('4279 Donec Road, Penza, TN', 'TRK597656582', 'UPS', '2025-04-11', '2025-04-11', 12.99),
108     ('4406 Velit. Rd, Belleville, KY', 'TRK147689627', 'USPS', '2025-10-22', '2025-10-23', 32.50);
109
110     -- Insert data into Products table
111 INSERT INTO Products (ProductID, ProductName, ProductDescription, Size, Color, Price) VALUES
112     ('PROD120000301', 'Cotton T-Shirt', 'Soft cotton t-shirt for everyday wear.', 'M', 'Blue', 25.00),
113     ('PROD232320102', 'Denim Jeans', 'Classic fit denim jeans.', '32x32', 'Indigo', 50.99),
114     ('PROD234340102', 'Denim Jeans', 'Classic fit denim jeans.', '34x34', 'Indigo', 50.99),
115     ('PROD130000064', 'Cotton T-Shirt', 'Soft cotton t-shirt for everyday wear.', 'L', 'Red', 25.00),
116     ('PROD130000223', 'Leather Jacket', 'Stylish Leather Jacket.', 'L', 'Brown', 150.99);
```

Figure 8

Online Apparel Store's Consumer Data Database Script Lines 117-145

```
117
118     -- Insert data into Orders table
119 •   INSERT INTO Orders (CustomerID, PaymentID, DeliveryID, OrderDate, OrderStatus, OrderTotalPrice) VALUES
120     (1, 1, 1, '2025-11-20 10:30:00', 'Delivered', 75.99),
121     (2, 2, 2, '2025-11-21 14:00:00', 'Processing', 120.50),
122     (3, 3, 3, '2025-01-28 20:50:03', 'Delivered', 17.40),
123     (4, 4, 4, '2025-04-07 23:10:46', 'Delivered', 18.95),
124     (5,5, 5, '2025-10-19 16:19:40', 'Delivered', 212.53);
125
126     -- Insert data into Reviews table
127 •   INSERT INTO Reviews (CustomerID, Rating, ReviewDate, ReviewText) VALUES
128     (2, 5, '2025-11-25 11:00:00', 'Great quality item, very comfortable!'),
129     (4, 4, '2025-11-29 09:00:00', 'Item fit well, but the color is slightly different than expected.'),
130     (5, 3, '2025-02-02 07:43:00', 'Comfortable, but a little expensive.'),
131     (2, 4, '2025-04-12 12:01:00', 'Great quality!'),
132     (1, 4, '2025-10-26 08:00:00', 'Great material, but fit a little tighter than expected.');
133
134     -- Insert data into Orders_Products junction table
135 •   INSERT INTO Orders_Products (OrderID, ProductID) VALUES
136     (1, 'PROD120000301'),
137     (2, 'PROD232320102'),
138     (3, 'PROD234340102'),
139     (4, 'PROD130000064'),
140     (5, 'PROD130000223');
141
142     -- Insert data into Reviews_Products junction table
143 •   INSERT INTO Reviews_Products (ReviewID, ProductID) VALUES
144     (1, 'PROD120000301'),
145     (2, 'PROD232320102'),
```

Figure 9

Online Apparel Store's Consumer Data Database Script Lines 146-148

```
146     (3, 'PROD234340102'),
147     (4, 'PROD120000301'),
148     (5, 'PROD130000223');
```

Table 1

Online Apparel Store's Consumer Data Database "AllTables" Table (part 1 of 4)

```
1 •   SELECT * FROM onlineapparelstoreconsumerdata.alltables;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CustomerID	FirstName	LastName	Email	PhoneNumber	PaymentID	PaymentType	PaymentDate	PaymentAmount	CardNum	DeliveryID	S
1	Alice	Smith	alice.smith@example.com	541-555-4567	1	Visa	2025-11-20 10:30:00	75.99	4432	1	12
2	Bob	Johnson	bob.j@example.com	817-555-6543	2	MasterCard	2025-11-21 14:00:00	120.50	6651	2	45
2	Bob	Johnson	bob.j@example.com	817-555-6543	2	MasterCard	2025-11-21 14:00:00	120.50	6651	2	45
3	Alex	Wayne	AWAYNE03@example.com	742-555-2418	3	Visa	2025-01-28 20:50:03	17.40	8642	3	P.
4	Myles	Keaton	Keaton64@example.com	732-555-7714	4	Discover	2025-04-07 23:10:46	18.95	9528	4	42
5	Laura	Hall	Laura.Hall641@example.com	791-555-6351	5	AmEx	2025-10-19 16:19:40	212.53	6458	5	44

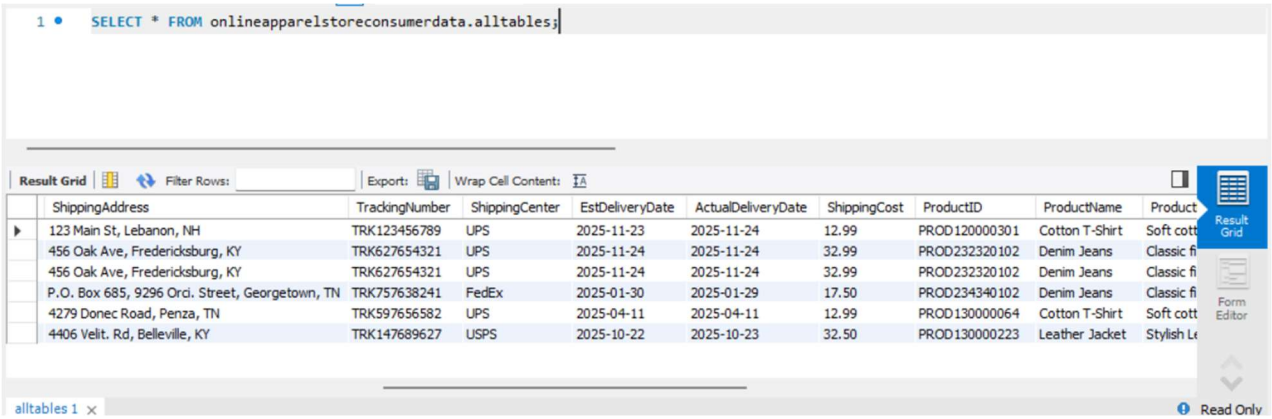
Result Grid | Form Editor

alltables 1 ×

Read Only

Table 2

Online Apparel Store's Consumer Data Database "AllTables" Table (part 2 of 4)



```
1 •  SELECT * FROM onlineapparelstoreconsumerdata.alltables;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

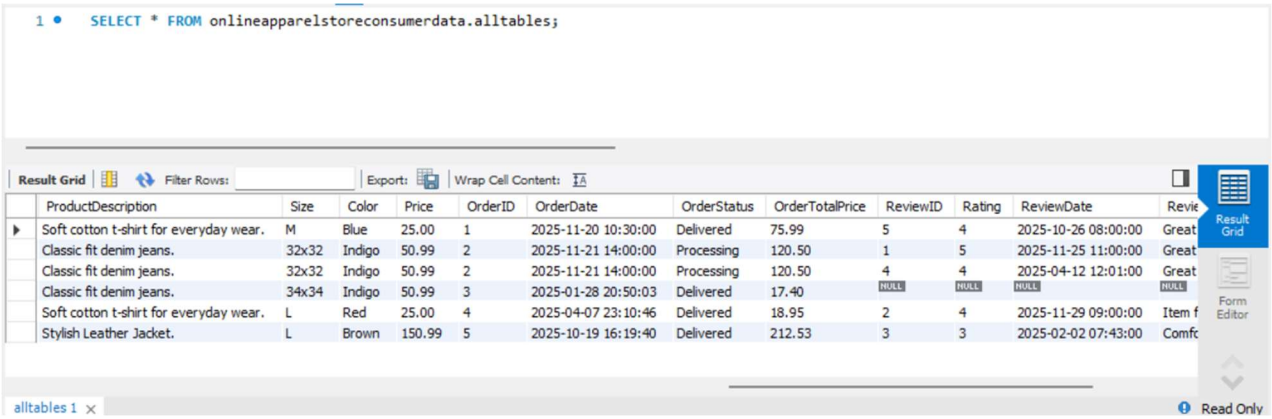
ShippingAddress	TrackingNumber	ShippingCenter	EstDeliveryDate	ActualDeliveryDate	ShippingCost	ProductID	ProductName	Product
123 Main St, Lebanon, NH	TRK123456789	UPS	2025-11-23	2025-11-24	12.99	PROD120000301	Cotton T-Shirt	Soft cott
456 Oak Ave, Fredericksburg, KY	TRK627654321	UPS	2025-11-24	2025-11-24	32.99	PROD232320102	Denim Jeans	Classic fi
456 Oak Ave, Fredericksburg, KY	TRK627654321	UPS	2025-11-24	2025-11-24	32.99	PROD232320102	Denim Jeans	Classic fi
P.O. Box 685, 9296 Orci. Street, Georgetown, TN	TRK757638241	FedEx	2025-01-30	2025-01-29	17.50	PROD234340102	Denim Jeans	Classic fi
4279 Donec Road, Penza, TN	TRK597656582	UPS	2025-04-11	2025-04-11	12.99	PROD130000064	Cotton T-Shirt	Soft cott
4406 Velit. Rd, Belleville, KY	TRK147689627	USPS	2025-10-22	2025-10-23	32.50	PROD130000223	Leather Jacket	Stylish Le

Result Grid | Form Editor

alltables 1 | Read Only

Table 3

Online Apparel Store's Consumer Data Database "AllTables" Table (part 3 of 4)



```
1 •  SELECT * FROM onlineapparelstoreconsumerdata.alltables;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

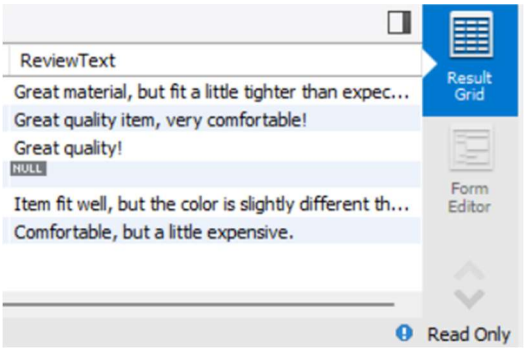
ProductDescription	Size	Color	Price	OrderID	OrderDate	OrderStatus	OrderTotalPrice	ReviewID	Rating	ReviewDate	Revie
Soft cotton t-shirt for everyday wear.	M	Blue	25.00	1	2025-11-20 10:30:00	Delivered	75.99	5	4	2025-10-26 08:00:00	Great
Classic fit denim jeans.	32x32	Indigo	50.99	2	2025-11-21 14:00:00	Processing	120.50	1	5	2025-11-25 11:00:00	Great
Classic fit denim jeans.	32x32	Indigo	50.99	2	2025-11-21 14:00:00	Processing	120.50	4	4	2025-04-12 12:01:00	Great
Classic fit denim jeans.	34x34	Indigo	50.99	3	2025-01-28 20:50:03	Delivered	17.40	NULL	NULL	NULL	NULL
Soft cotton t-shirt for everyday wear.	L	Red	25.00	4	2025-04-07 23:10:46	Delivered	18.95	2	4	2025-11-29 09:00:00	Item f
Stylish Leather Jacket.	L	Brown	150.99	5	2025-10-19 16:19:40	Delivered	212.53	3	3	2025-02-02 07:43:00	Comfc

Result Grid | Form Editor

alltables 1 | Read Only

Table 4

Online Apparel Store's Consumer Data Database "AllTables" Table (part 4 of 4)



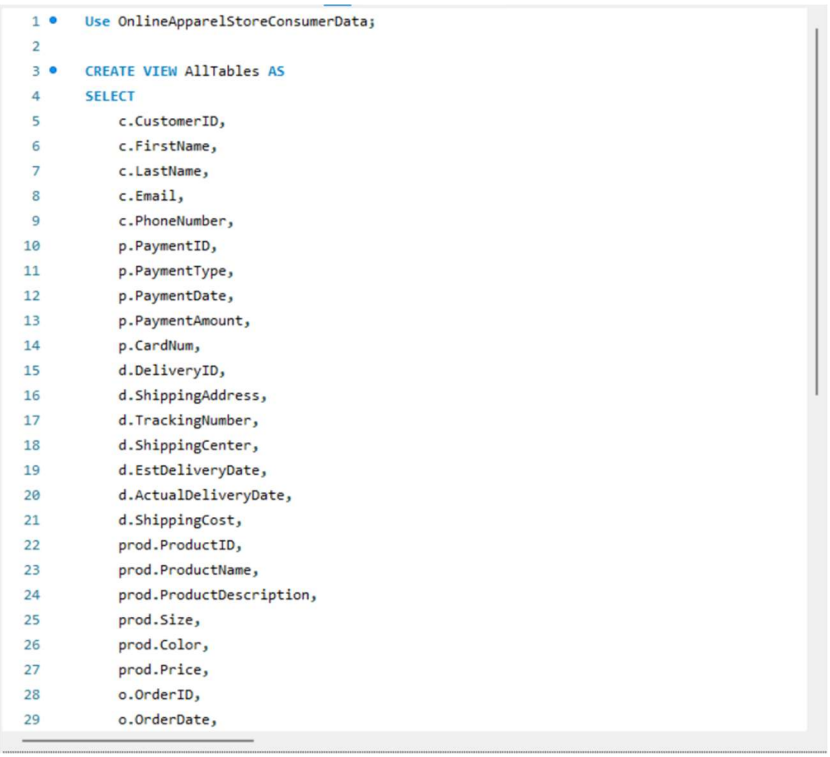
ReviewText
Great material, but fit a little tighter than expec...
Great quality item, very comfortable!
Great quality!
NULL
Item fit well, but the color is slightly different th...
Comfortable, but a little expensive.

Result Grid | Form Editor

Read Only

Figure 10

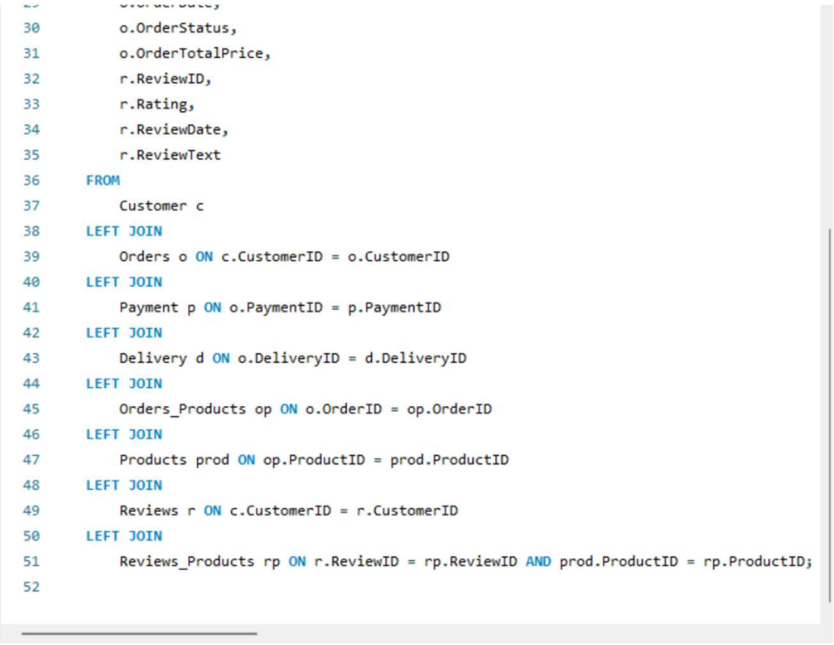
Online Apparel Store's Consumer Data Database "AllTables" Script Lines 1-29



```
1  Use OnlineApparelStoreConsumerData;
2
3  CREATE VIEW AllTables AS
4  SELECT
5      c.CustomerID,
6      c.FirstName,
7      c.LastName,
8      c.Email,
9      c.PhoneNumber,
10     p.PaymentID,
11     p.PaymentType,
12     p.PaymentDate,
13     p.PaymentAmount,
14     p.CardNum,
15     d.DeliveryID,
16     d.ShippingAddress,
17     d.TrackingNumber,
18     d.ShippingCenter,
19     d.EstDeliveryDate,
20     d.ActualDeliveryDate,
21     d.ShippingCost,
22     prod.ProductID,
23     prod.ProductName,
24     prod.ProductDescription,
25     prod.Size,
26     prod.Color,
27     prod.Price,
28     o.OrderID,
29     o.OrderDate,
```

Figure 11

Online Apparel Store's Consumer Data Database "AllTables" Script Lines 30-52



```
30     o.OrderStatus,
31     o.OrderTotalPrice,
32     r.ReviewID,
33     r.Rating,
34     r.ReviewDate,
35     r.ReviewText
36 FROM
37     Customer c
38 LEFT JOIN
39     Orders o ON c.CustomerID = o.CustomerID
40 LEFT JOIN
41     Payment p ON o.PaymentID = p.PaymentID
42 LEFT JOIN
43     Delivery d ON o.DeliveryID = d.DeliveryID
44 LEFT JOIN
45     Orders_Products op ON o.OrderID = op.OrderID
46 LEFT JOIN
47     Products prod ON op.ProductID = prod.ProductID
48 LEFT JOIN
49     Reviews r ON c.CustomerID = r.CustomerID
50 LEFT JOIN
51     Reviews_Products rp ON r.ReviewID = rp.ReviewID AND prod.ProductID = rp.ProductID;
52
```