Anthony Upchurch

ITSE 2309

Mastery Assessment M6A

*Resubmitted with corrections*

**Real-world Application:** Currently, I am an office manager for a wholesaler. We often must process Return Merchandise Agreements (RMA) for customers who receive damaged/defective items. Processing the RMA requires two separate systems, Refund Tracker and Returned Merchandise Tracker. The creation of an Inventory Returns database would allow us to streamline the process of tracking both the RMAs and the refunds, which would allow us to identify trends in product defects and track profit/loss margins in the production of our products.

**Tables Required:** Customer Order Link Table (Bridging Table), Customers, Inventory Table, RMAs Table, Orders Table, Order Items Table, Returns Table. Primary Keys appear in Green, Foreign Keys appear in Purple.

|  |  |  |
| --- | --- | --- |
|  | **Customer\_Order\_Link** |  |
| **LinkID** | **Email** | **CustomerID** |
| 1 | anthony@americandarlingbag.com | 1 |
| 2 | janedoeboutique@gmail.com | 2 |
| 3 | thefadedbutterflyco@gmail.com | 3 |
| 4 | hideandfindge@yahoo.com | 4 |
| 5 | theturquoiseflamingo@gmail.com | 5 |

The Customer\_Order\_Link Table will act as a *bridging table* between the current sales system and the new Inventory Returns database. Because the sales system uses the customer’s email address to identify the entry, I would need a link between the two databases to populate the information I need. The sales system uses the customer’s email address as a Foreign Key to pass information to the invoicing system and the shipping system. Unfortunately, there have been situations where a customer changed their address, which causes orders to become “lost” in the system. This flaw will carry into my new database due to the relationship the email address plays in the transfer of data between the databases currently being used.

|  |  |
| --- | --- |
| **Customers** | |
| **CustomerID** | **OrderID** |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |

The **Customers Table** will allow me to link the customer’s information directly from the sales system via the CustomerOrderLink Table. This allows me to minimize the need to enter additional data and utilize the database that is already in place in a more effective way. The CustomerID in the CustomerOrderLink Table acts as a Foreign Key to bring the data into the Customers Table. It will also allow me to populate the data of each order in the sales system which will be needed for this database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Orders** | |  |
| **OrderID** | **CustomerID** | **OrderNumber** | **PaymentType** | **TotalAmount** |
| 1 | 1 | ADN210034 | Shopify | 365.00 |
| 2 | 2 | ADN210057 | PayPal | 1254.50 |
| 3 | 3 | ADN210086 | Shopify | 11254.00 |
| 4 | 4 | ADN210125 | AC\_MERCHANT | 542.50 |
| 5 | 5 | ADN210158 | NET30 | 789.00 |

With the OrderID I can store the information for the orders passed from the sale system and link it to the customer by the CustomerID. This will allow me to determine the tender used by a customer when processing any refunds later, it also allows me to link the order number to the customer for processing in other tables of the database.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Order\_Items** | |  |
| **OrderItemID** | **OrderID** | **SKU** | **QuantityOrdered** |
| 1 | 1 | ADBG1234 | 3 |
| 2 | 1 | ADBG1236 | 4 |
| 3 | 3 | ADBG1237 | 5 |
| 4 | 3 | ADBG1238 | 7 |
| 5 | 5 | ADBG1239 | 1 |

The Order\_Items Table provides a way to track the items being returned and reference the total quantity order for each item being refunded. This can give the database user insight into the number of damaged or defective items being returned vs. the amount ordered. This could be used to determine if the entire case of the item might be damaged or just the one being returned. A query could be utilized to provide the percentage of QuantityReturned vs. QuantityOrdered to determine if warehouse inventory needs to be submitted for a Quality Assurance check by the Inventory Team.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Inventory** | |  |
| **SKU** | **WholesaleCost** | **ListCost** | **ProfitAmount** |
| ADBG1234 | 21.10 | 1.33 | 19.77 |
| ADBG1236 | 16.00 | 0.80 | 15.20 |
| ADBG1237 | 33.00 | 2.20 | 30.80 |
| ADBG1238 | 32.00 | 1.25 | 30.75 |
| ADBG1239 | 21.80 | 0.51 | 21.29 |

The Inventory Table is a *bridging table* which will store cost related data obtained from an outside inventory database managed by the SKUVault program. This will allow the database user to provide detailed reports containing the manufacture and cost of the product being returned and outline the amount of profit lost due to the item being returned. This can maximize the efficiency of the Claims Department and give upper management a better ability to track company financial situations.

|  |  |  |
| --- | --- | --- |
| **RMA** | | |
| **RMAID** | **RMANumber** | **OrderNumber** |
| 1 | 102801 | ADN210034 |
| 2 | 102802 | ADN210057 |
| 3 | 102803 | ADN210086 |
| 4 | 102804 | ADN210125 |
| 5 | 102805 | ADN210158 |

The RMA Tablestores the information about the RMA and links the RMAs to the customer and the order data. This could allow the database user to track information such as complete and pending RMAs, customers who have repeated RMA entries and whether orders contain multiple RMAs. This gives the database user the ability to identify potential fraud issues and also insures higher efficiency in the RMA process.

|  |  |  |  |
| --- | --- | --- | --- |
| **RMA Items** | | | |
| **RMAItemID** | **RMAID** | **SKU** | **Quantity** |
| 1 | 1 | ADBG1234 | 1 |
| 2 | 1 | ADBG1236 | 2 |
| 3 | 3 | ADBG1237 | 1 |
| 4 | 3 | ADBG1238 | 1 |
| 5 | 5 | ADBG1239 | 4 |

The RMA\_Items Table provides a way to store the items contained in an RMA request. Because there can be multiple items returned under one RMA, it is important to have the ability to track each of the items contained in the RMA. The table also links the SKU number in the RMA request to the Inventory Table which allows the database to retrieve data faster through table relationships.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Returns** |  | | |
| **ReturnID** | **RMAID** | **SKU** | **QuantityReturned** | | **ActionsTaken** | **DateReturned** |
| 1 | 1 | ADBG1234 | 1 | | Refund | 2024-09-01 |
| 2 | 1 | ADBG1236 | 2 | | Refund | 2024-09-01 |
| 3 | 3 | ADBG1237 | 1 | | Exchange | 2024-10-22 |
| 4 | 3 | ADBG1238 | 1 | | Refund | 2024-10-22 |
| 5 | 5 | ADBG1239 | 4 | | Exchange | 2024-11-21 |

The Returns Table is used to store detailed information about the returns or exchanges being processed by the office. It not only would allow for detailed financial reporting but also provides a way to link the returns to both the RMA table and the Inventory Table. This would allow the database user to query information about the orders that require refunds and exchanges, and again could be used to determine if there is an issue within the inventory of the warehouse.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Returned\_Merchandise** | | | | | | | |
| **ClaimID** | **ReturnID** | **SKU** | **Supplier** | **Quantity** | **ListCost** | **ProfitAmount** | **AmountOfLoss** |
| 1 | 1 | ADBG1234 | GIN | 1 | 1.33 | 19.77 | 19.77 |
| 2 | 1 | ADBG1236 | GIN | 2 | 0.8 | 15.2 | 30.4 |
| 3 | 3 | ADBG1237 | VARNO | 1 | 2.2 | 30.8 | 30.8 |
| 4 | 3 | ADBG1238 | GIN | 1 | 1.25 | 30.75 | 30.75 |
| 5 | 5 | ADBG1239 | HILASON | 4 | 0.51 | 21.29 | 85.16 |

The Returned\_Merchandise Table links the RMA data and the Inventory data to create a table for claims processing. This table would be primarily used to generate various reports relating to claims processing and production management. However, it can also be used by the office to determine if a work ticket should be issued to the Inventory Department to check for damaged or defective shipments from the supplier. Minimizing the over loss to the company from the shipping cost involved in retrieving and exchanging damaged or defective items received by the customer.