Information Management Final Project

Information Management

Professor Yifan Yu

Jagruta Advani, Dhruv Arora, Rohan Giri, Sonali Hornick, Deeksha Koonadi

01 December 2024

Information Management Report

Jagruta Advani, Dhruv Arora, Rohan Giri, Sonali Hornick, Deeksha Koonadi

MS in Business Analytics

TARGET Data Models

In our report for Target, we assume these key Transaction Management Systems and Their Data Structures mentioned below :

1. **Sales and Payment Processing System**
2. **Customer Loyalty and Engagement System**
3. **Order Fulfillment and Inventory Management System**

---

1. Sales and Payment Processing System

Data Model

The Sales and Payment Processing System manages customer orders, payments, and invoices. Below is the data model:

1. **Customer**: Stores customer information such as name, contact details, and address.
2. **SalesOrder**: Tracks customer orders, including order dates and statuses.
3. **SalesOrderItem**: Links products to specific sales orders, with details about quantities and prices.
4. **Product**: Stores product details such as name, description, price, and stock.
5. **Payment**: Tracks payment details for each sales order, including payment methods, dates, and amounts.
6. **PaymentMethod**: Manages various payment options (e.g., credit card, cash).
7. **Invoice**: Tracks invoices generated for completed sales orders.

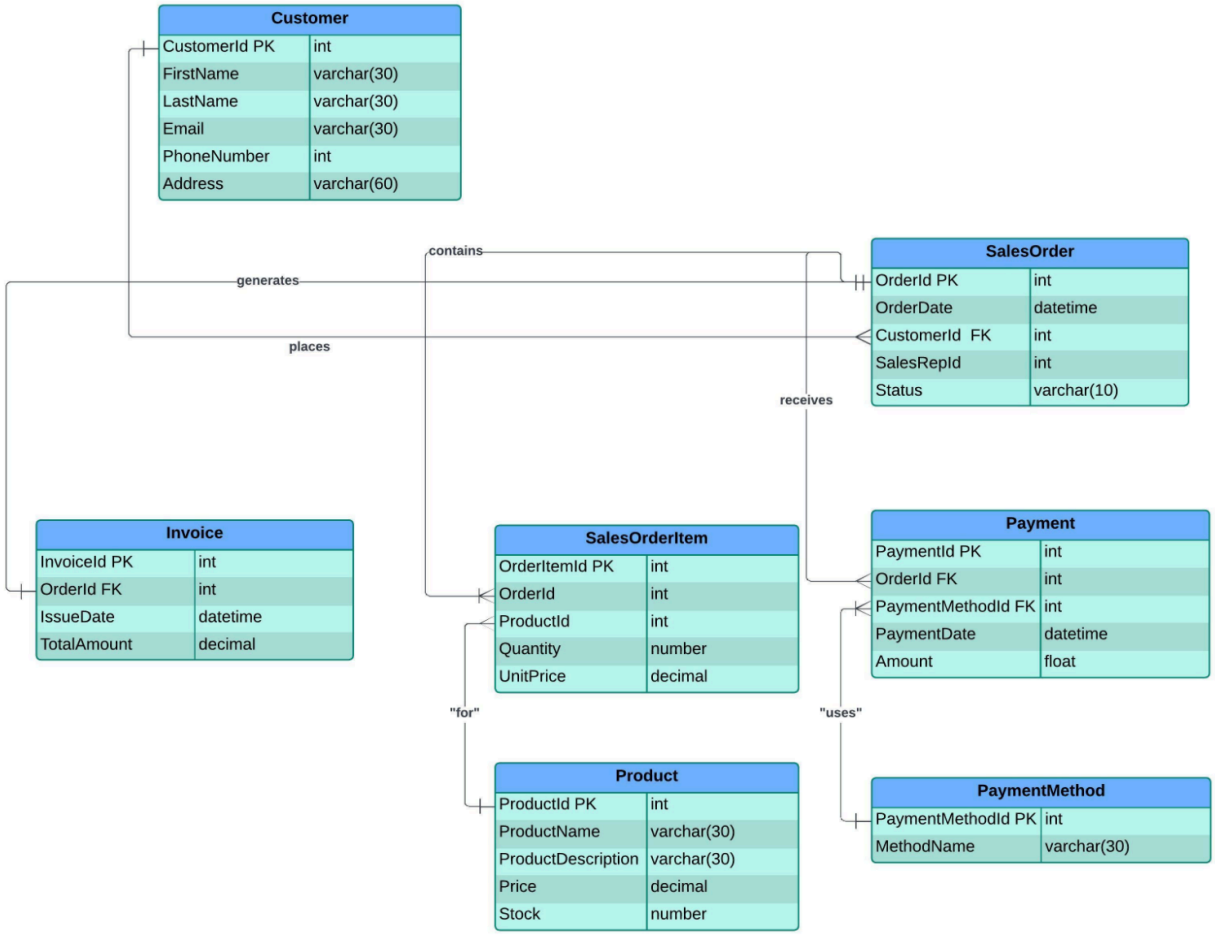
```
CREATE TABLE "Product" (
  "ProductId PK" int,
  "ProductName" varchar(30),
  "ProductDescription" varchar(30),
  "Price" decimal,
  "Stock" number
);
```

```
CREATE TABLE "PaymentMethod" (
  "PaymentMethodId PK" int,
  "MethodName" varchar(30)
);

CREATE TABLE "Customer" (
  "CustomerId PK" int,
  "FirstName" varchar(30),
  "LastName" varchar(30),
  "Email" varchar(30),
  "PhoneNumber" int,
  "Address" varchar(60)
);

CREATE TABLE "SalesOrder" (
  "OrderId PK" int,
  "OrderDate" datetime,
  "CustomerId  FK" int,
  "SalesRepId" int,
  "Status" varchar(10),
  CONSTRAINT "FK_SalesOrder.CustomerId  FK"
    FOREIGN KEY ("CustomerId  FK")
      REFERENCES "Customer"("CustomerId PK")
);
```

```
CREATE TABLE "Payment" (
  "PaymentId PK" int,
  "OrderId FK" int,
  "PaymentMethodId FK" int,
  "PaymentDate" datetime,
  "Amount" float,
  CONSTRAINT "FK_Payment.PaymentMethodId FK"
    FOREIGN KEY ("PaymentMethodId FK")
      REFERENCES "PaymentMethod"("PaymentMethodId PK"),
  CONSTRAINT "FK_Payment.OrderId FK"
    FOREIGN KEY ("OrderId FK")
      REFERENCES "SalesOrder"("OrderId PK")
);
CREATE TABLE "SalesOrderItem" (
  "OrderItemId PK" int,
  "OrderId" int,
  "ProductId" int,
  "Quantity" number,
  "UnitPrice" decimal,
  CONSTRAINT "FK_SalesOrderItem.ProductId"
    FOREIGN KEY ("ProductId")
      REFERENCES "Product"("ProductId PK"),
  CONSTRAINT "FK_SalesOrderItem.OrderId"
    FOREIGN KEY ("OrderId")
      REFERENCES "SalesOrder"("OrderId PK")
);
```

```
CREATE TABLE "Invoice" (
  "InvoiceId PK" int,
  "OrderId FK" int,
  "IssueDate" datetime,
  "TotalAmount" decimal
);
```

Sales and Payment Processing System Data Model



2. Customer Loyalty and Engagement System

Data Model

The Customer Loyalty and Engagement System focuses on managing customer relationships, tracking loyalty rewards, capturing feedback, and recording preferences to enhance customer engagement. Below is the data model:

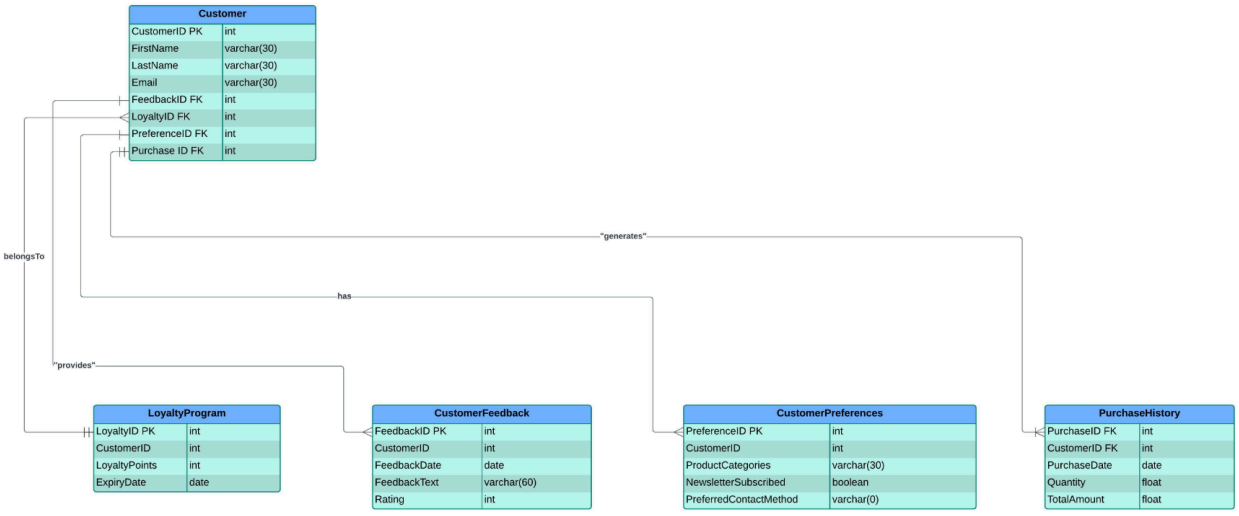
1. **LoyaltyProgram:** Tracks customer loyalty details, including points earned, points redeemed, and expiration dates.
2. **Customer:** Stores customer personal information and links to feedback, preferences, purchase history, and loyalty programs.
3. **PurchaseHistory:** Maintains a record of customer purchases, including the purchase date, quantity, and total amount spent.
4. **Customer Feedback:** Captures feedback provided by customers, including ratings and comments.
5. **Customer preferences:** Records customer preferences for products, communication methods, and subscription to newsletters.

```
CREATE TABLE "LoyaltyProgram" (
  "LoyaltyID PK" int,
  "CustomerID" int,
  "LoyaltyPoints" int,
  "ExpiryDate" date
);
```

```
CREATE TABLE "Customer" (
  "CustomerID PK" int,
  "FirstName" varchar(30),
  "LastName" varchar(30),
  "Email" varchar(30),
  "FeedbackID FK" int,
  "LoyaltyID FK" int,
  "PreferenceID FK" int,
  "Purchase ID FK" int,
  CONSTRAINT "FK_Customer.LoyaltyID FK"
    FOREIGN KEY ("LoyaltyID FK")
      REFERENCES "LoyaltyProgram"("LoyaltyID PK")
);

CREATE TABLE "PurchaseHistory" (
  "PurchaseID FK" int,
  "CustomerID FK" int,
  "PurchaseDate" date,
  "Quantity" float,
  "TotalAmount" float,
  CONSTRAINT "FK_PurchaseHistory.PurchaseID FK"
    FOREIGN KEY ("PurchaseID FK")
      REFERENCES "Customer"("Purchase ID FK")
```

```
);

CREATE TABLE "CustomerFeedback" (
  "FeedbackID PK" int,
  "CustomerID" int,
  "FeedbackDate" date,
  "FeedbackText" varchar(60),
  "Rating" int,
  CONSTRAINT "FK_CustomerFeedback.FeedbackID PK"
    FOREIGN KEY ("FeedbackID PK")
      REFERENCES "Customer"("FeedbackID FK")
);

CREATE TABLE "CustomerPreferences" (
  "PreferenceID PK" int,
  "CustomerID" int,
  "ProductCategories" varchar(30),
  "NewsletterSubscribed" boolean,
  "PreferredContactMethod" varchar(0),
  CONSTRAINT "FK_CustomerPreferences.PreferenceID PK"
    FOREIGN KEY ("PreferenceID PK")
      REFERENCES "Customer"("PreferenceID FK")
);
```

Customer Loyalty and Engagement System Model



3. Order Fulfillment and Inventory Management System

Data Model

The Order Fulfillment and Inventory Management System is designed to manage inventory levels, process customer orders, and handle shipments. Below is the data model:

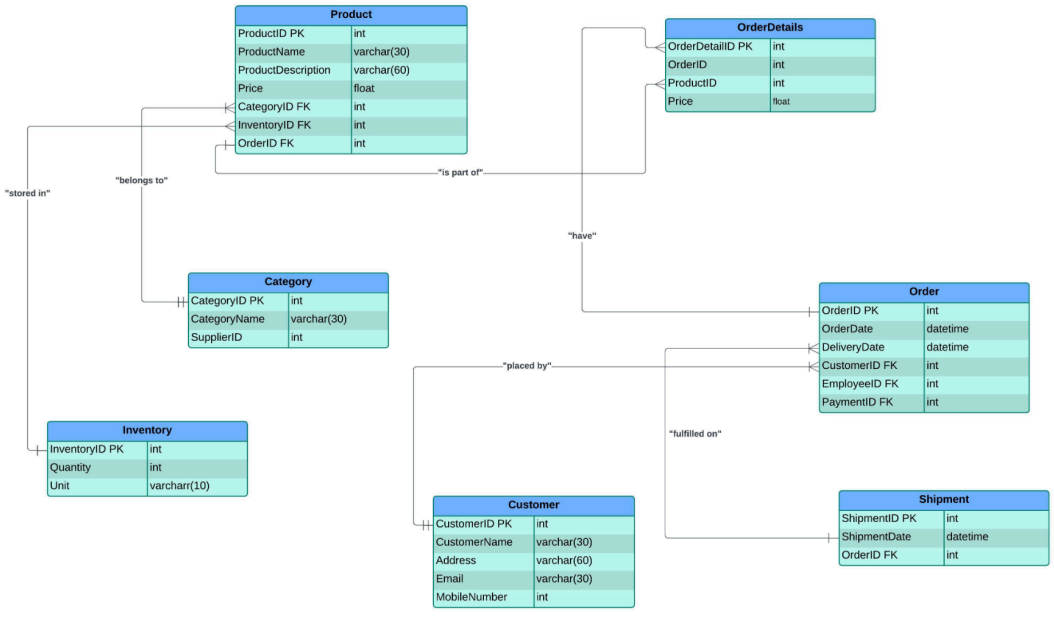
1. **Customer:** Stores customer information, including name, contact details, and address.
2. **Order:** Tracks customer orders, including order date, delivery date, and associated details.
3. **Order Details:** Links products to specific orders, capturing product details and pricing.
4. **Product:** Stores product information, such as name, description, price, and associated categories.
5. **Inventory:** Manages stock levels, units, and warehouse locations for products.
6. **Category:** Organizes products into categories and associates them with suppliers.
7. **Shipment:** Tracks details of shipments, including shipment date, associated orders, and locations.

```
CREATE TABLE "Customer" (
  "CustomerID PK" int,
  "CustomerName" varchar(30),
  "Address" varchar(60),
  "Email" varchar(30),
  "MobileNumber" int
);
CREATE TABLE "Shipment" (
  "ShipmentID PK" int,
  "ShipmentDate" datetime,
  "OrderID FK" int
);
CREATE TABLE "Inventory" (
  "InventoryID PK" int,
  "Quantity" int,
  "Unit" varcharr(10)
);
CREATE TABLE "Order" (
  "OrderID PK" int,
  "OrderDate" datetime,
  "DeliveryDate" datetime,
  "CustomerID FK" int,
  "EmployeeID FK" int,
  "PaymentID FK" int,
  CONSTRAINT "FK_Order.CustomerID FK"
    FOREIGN KEY ("CustomerID FK")
```

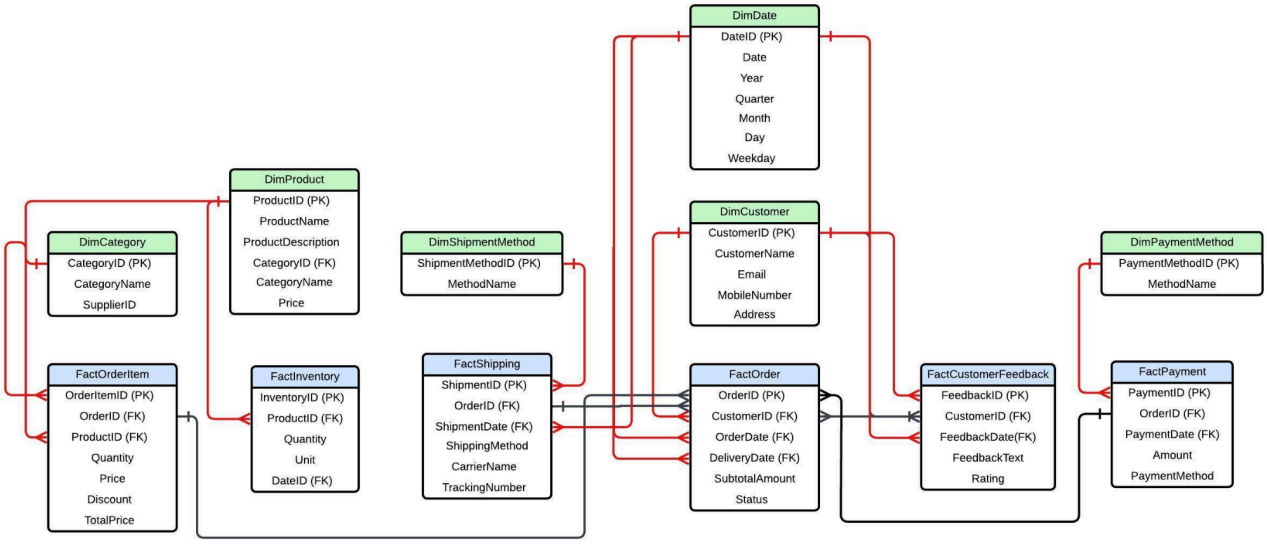
```
    REFERENCES "Customer"("CustomerID PK"),
  CONSTRAINT "FK_Order.DeliveryDate"
    FOREIGN KEY ("DeliveryDate")
      REFERENCES "Shipment"("ShipmentDate")
);
CREATE TABLE "Category" (
  "CategoryID PK" int,
  "CategoryName" varchar(30),
  "SupplierID" int
);
CREATE TABLE "Product" (
  "ProductID PK" int,
  "ProductName" varchar(30),
  "ProductDescription" varchar(60),
  "Price" float,
  "CategoryID FK" int,
  "InventoryID FK" int,
  "OrderID FK" int,
  CONSTRAINT "FK_Product.CategoryID FK"
    FOREIGN KEY ("CategoryID FK")
      REFERENCES "Category"("CategoryID PK"),
  CONSTRAINT "FK_Product.InventoryID FK"
    FOREIGN KEY ("InventoryID FK")
      REFERENCES "Inventory"("InventoryID PK")
);
CREATE TABLE "OrderDetails" (
```

```
  "OrderDetailID PK" int,
  "OrderID" int,
  "ProductID" int,
  "Price" float,
  CONSTRAINT "FK_OrderDetails.ProductID"
    FOREIGN KEY ("ProductID")
      REFERENCES "Product"("OrderID FK"),
  CONSTRAINT "FK_OrderDetails.OrderDetailID PK"
    FOREIGN KEY ("OrderDetailID PK")
      REFERENCES "Order"("OrderID PK")
);
```

Order Fulfillment and Inventory Management System



Data Warehouse



The enterprise data warehouse should seamlessly integrate these entities while adhering to normalization principles to ensure data consistency and integrity. The key entities and their relationships in the data warehouse model are structured as follows:

A. **Fact Tables:** Central repositories for quantitative data, capturing transactional and measurable events.
B. **Dimension Tables:** Supporting structures providing descriptive attributes to contextualize the data in fact tables.

Fact Tables

1. **FactOrderItem**:
    - Captures product-level details for each order.
    - **Columns**:
        - OrderItemID (PK)
        - OrderID (FK)
        - ProductID (FK)

        - QuantityOrder
        - Price
        - Discount
        - TotalPrice

2. **FactOrder**:
    - Tracks details about customer orders, such as total amount, tax, and status.
    - **Columns**:
        - OrderID (PK)
        - CustomerID (FK)
        - OrderDate (DateID FK)
        - SubtotalAmount
        - TaxAmount
        - InvoiceTotalAmount
        - Status

3. **FactInventory**:
    - Tracks product inventory levels in warehouses.
    - **Columns**:
        - InventoryID (PK)
        - ProductID (FK)
        - WarehouseID (FK)
        - Quantity
        - Unit
        - DateID (FK)

4. **FactPayment**:
    - Records payment transactions.
    - **Columns**:
        - PaymentID (PK)
        - OrderID (FK)
        - PaymentDate (DateID FK)

        - Amount
        - PaymentMethodID (FK)
        - PaymentStatus

5. **FactCustomerFeedback**:
    - Stores customer feedback on products or orders.
    - **Columns**:
        - FeedbackID (PK)
        - CustomerID (FK)
        - FeedbackDate (DateID FK)
        - FeedbackText
        - Rating

6. **FactShipping**:
    - Tracks shipping details, including carrier and delivery status.
    - **Columns**:
        - ShippingID (PK)
        - OrderID (FK)
        - ShippingDate (DateID FK)
        - FulfillmentType
        - ShippingMethod
        - CarrierName
        - TrackingNumber
        - EstimatedDeliveryDate (DateID FK)

Dimension Tables

1. **DimCustomer**:
    - Stores customer details for personalized analytics.
    - **Columns**:
        - CustomerID (PK)
        - CustomerFirstName

        - CustomerLastName
        - Email
        - MobileNumber
        - Address
2. **DimProduct**:
    - Stores product details, including pricing and categories.
    - **Columns**:
        - ProductID (PK)
        - ProductName
        - ProductDescription
        - CategoryID (FK)
        - Price
        - IsActive
3. **DimCategory**:
    - Organizes products into categories and links them to suppliers.
    - **Columns**:
        - CategoryID (PK)
        - CategoryName
        - SupplierID (FK)
4. **DimWarehouse**:
    - Tracks warehouse locations and capacities.
    - **Columns**:
        - WarehouseID (PK)
        - WarehouseName
        - LocationID (FK)
5. **DimLocation**:
    - Stores detailed location data for warehouses and shipping.
    - **Columns**:
        - LocationID (PK)
        - Address
        - City
        - State
        - Country
        - PostalCode

6. **DimDate**:
    - Provides a time dimension for analytics.
    - **Columns**:
        - DateID (PK)
        - Date
        - Year
        - Quarter
        - Month
        - Day
        - Weekday
7. **DimPaymentMethod**:
    - Stores various payment methods used by customers.
    - **Columns**:
        - PaymentMethodID (PK)
        - MethodName
8. **DimShipmentMethod**:
    - Stores shipment methods, such as delivery type or carrier details.
    - **Columns**:
        - ShipmentMethodID (PK)
        - MethodName