E-Commerce Case Study: Database Design and Implementation

## Written by :-

## Arshdeep singh dhaliwal

**Table of Contents**

1. Introduction
2. Mission Statement
3. Business Rules
4. Objectives
5. List of Tables
6. List of Attributes
7. Entity-Relationship Diagram
8. Data Dictionary
9. Views
10. Conclusion

# 1. Introduction

This case study aims at designing a database system for an e-commerce platform. The system is to efficiently handle customer information, employee data, product inventory, order and shipment information, supplier, and payment-related details. In short, it is to assist business operations through better decision-making by systematically storing and retrieving data

# 2. Mission Statement

The goal is to design a well-structured and efficient database system that will support e-commerce operations by maintaining accurate records, ensuring smooth transactions, and facilitating data-driven insights for business growth..

# 3. Business Rules

**Customer Rules:**• There should be a unique identifier for each customer.  
• The customer record should be maintained along with personal and contact information. **Employee Rules:**• Every employee should have a Unique Identifier.  
• Employee details include name, designation, skill, availability, and salary. **Supplier Rules:**• Every supplier should have a Unique Identifier.  
• Supplier details should include contact and details about the products they supply.

Order Rules:• Every order should have a Unique Identifier.  
• Order will be related to a customer and it should contain product details, shipment, andpayment details**.  
Product Rules:  
•**There shall be an ID for every product.  
• The products should include descriptions, quantity in stock, and price. Payment Rules:  
• The payments shall be made against an order.  
• Payment information includes method, reference ID, and total amount

**. Shipment Rules**:  
• The shipments must have a unique tracking number.  
• The shipment status shall be marked as "In Transit," "Delivered," or "Returned."4.

# 4 OBJECTIVES

1. **Efficient Order & Customer Management** → Tracks **customer purchases**, **order history**, and **transactions**.
2. **Organized Inventory & Supplier Tracking** → Ensures accurate **stock levels** and **supplier information**.
3. **Reliable Logistics & Customer Engagement** → Tracks **shipments, payments, and customer preferences** for better service.

**How does the database achieve this?**

* By organizing data in separate tables (Customers, Products, Orders, etc.), Trendify ensures **efficient data retrieval**.

This eliminates duplicate data, reduces errors, and **optimizes e-commerce operations**.

# 5. List of Tables

|  |
| --- |
| 1. Customers |
| 1. Employees |
| 1. Suppliers |
| 1. Products |
| 1. Orders |
| 1. Payment Methods |
| 1. Shipments |
| 1. Offers |

# 6. List of Attributes& constraints

**Customer Table:**

|  |  |
| --- | --- |
| Customer\_ID | Primary Key) |
| First\_Name |  |
| Last\_Name |  |
| Email |  |
| Phone |  |
| Address (House Number, Street, City, Province, Zip Code) |  |

**Employee Table:**

|  |  |
| --- | --- |
| Employee\_ID | (Primary Key) |
| Name |  |
| Role |  |
| Skills |  |
| Availability |  |
| Hire Date |  |
| Resign Date |  |
| Salary |  |

**Product Table:**

|  |  |
| --- | --- |
| Product\_ID | (Primary Key) |
| Product\_Name |  |
| Description |  |
| Quantity\_In\_Stock |  |
| Price |  |
| Supplier\_ID | (Foreign Key) |

**Order Table:**

|  |  |
| --- | --- |
| Order\_ID | (Primary Key) |
| Customer\_ID | (Foreign Key) |
| Order\_Date |  |
| Quantity |  |
| Product\_ID | (Foreign Key) |
| Employee\_ID | (Foreign Key) |
| Shipment\_ID | (Foreign Key) |
| Offer\_ID | (Foreign Key) |

**Shipment Table:**

|  |  |
| --- | --- |
| Shipment\_ID | (Primary Key) |
| Shipment\_Date |  |
| Tracking\_Number |  |
| Status (In Transit, Delivered, Returned) |  |

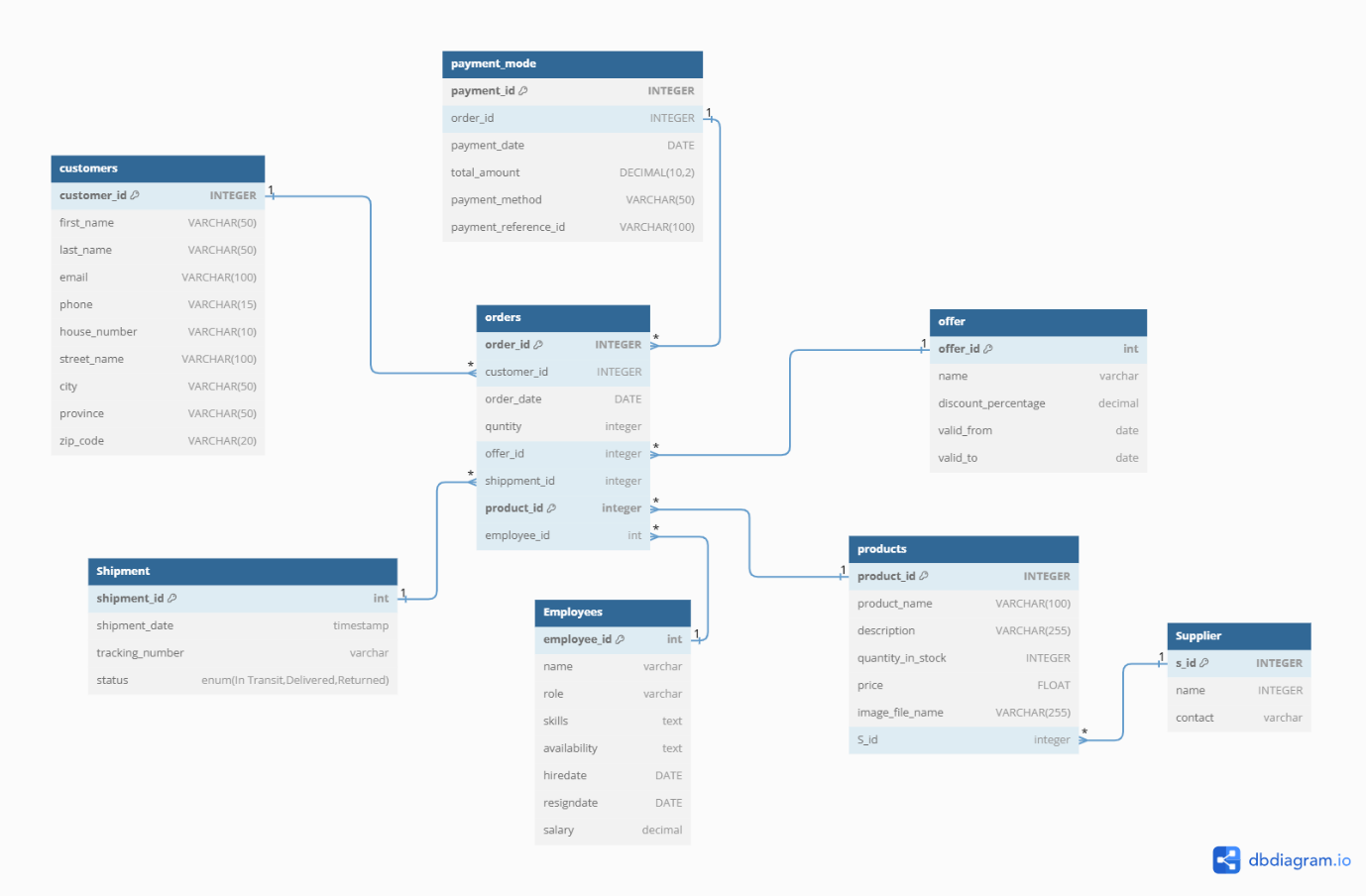
**Payment Table:**

|  |  |
| --- | --- |
| Payment\_ID | (Primary Key) |
| Order\_ID | (Foreign Key) |
| Payment\_Date |  |
| Total\_Amount |  |
| Payment\_Method |  |
| Payment\_Reference\_ID |  |

**Offer Table:**

|  |  |
| --- | --- |
| Offer\_ID | (Primary Key) |
| Name |  |
| Discount\_Percentage |  |
| Valid\_From |  |
| Valid\_To |  |
|  |  |

# Er diagram :-

****

# 7. Entity-Relationship Diagram

* The ER diagram for this e-commerce database represents key relationships,
* including: Customers ↔ Orders (One-to-Many)
  + One customer can place multiple orders, but each order belongs to only one customer.
* Orders → Products (Many-to-One)
  + Every order **must include at least one product.**
* Orders ↔ Payments (One-to-One)
  + Each order has one payment transaction
* Orders → Shipments (One-to-One)
  + Each order has one shipment record with tracking details.
* Orders → Offers (Many-to-One)
  + Customers can receive **discounts on their orders**.
* Products → Suppliers (Many-to-One)
  + Every **product is supplied by a vendor**.
* Orders → Employees (Many-to-One)
  + Each order is assigned to an employee for management.

# 8. Data Dictionary

**Customers Table:**

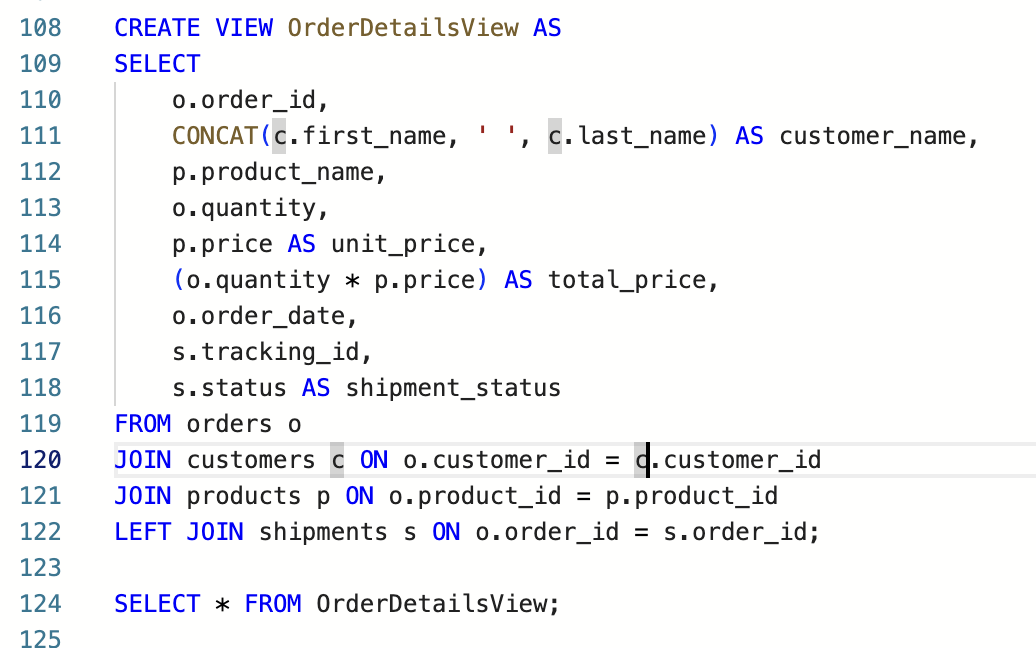
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  |  |  |  |  |   **Field** | **Key** | **Data Type** |
| Customer\_ID | Primary Key | Integer |
| First\_Name |  | Varchar(50) |
| Last\_Name |  | Varchar(50) |
| Email |  | Varchar(100) |
| Phone |  | Varchar(15) |
| House\_Number |  | Varchar(10) |
| Street\_Name |  | Varchar(100) |
| City |  | Varchar(50) |
| Province |  | Varchar(50) |
| Zip\_Code |  | Varchar(20) |

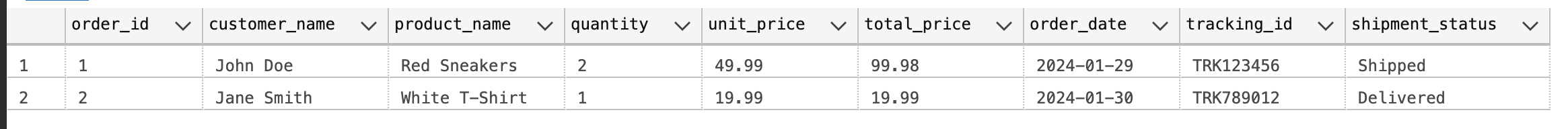
**Orders Table:**

|  |  |  |
| --- | --- | --- |
| Field | Key | Data Type |
| Order\_ID | Primary Key | Integer |
| Customer\_ID | Foreign Key | Integer |
| Order\_Date |  | Date |
| Quantity |  | Integer |
| Offer\_ID | Foreign Key | Integer |
| Shipment\_ID | Foreign Key | Integer |
| Product\_ID | Foreign Key | Integer |
| Employee\_ID | Foreign Key | Integer |

# 9 .views

# Order Details VieW:- To gather detailed order information, providing insights into customer purchases, pricing, and transaction history

****

****

# Invoice view:- To gather information for tracking invoices per customer by including customer details, order items, and payment transactions

# Inventory view: To gather real-time inventory data, for accurate tracking of stock levels, supplier sources, and pricing.

# 10. Conclusion

This database design ensures an efficient and organized e-commerce platform by integrating customer, product, order, and payment management. By leveraging this structured approach, businesses can improve decision-making, optimize inventory, streamline order processing, and enhance customer satisfaction. The well-defined relationships between entities provide a scalable solution for future growth and technological advancements.