P2: Database Design and Initial ERD

Group 7 – Minju Kim, Yue Liu, Ali Lo, Jess Madson, Natasha Sumarta

Database Topic – Online Retail Database (Women's Clothes & Accessories)

Database Purpose

The purpose of this database is to maintain the online shopping data used by a women's clothing and accessories store to track information about and related to sales, inventory/suppliers, customers, and products. It will be used by retail, warehouse, and customer staff.

Business Problems Addressed

- Provide customer-specific information related to buying history (e.g., sales, returns), order amounts, contact information, and payment methods to better understand customer needs
- Understand customer sentiment as it relates to specific products via reviews (e.g., text, Likert scale)
- Supply chain management to maintain the warehouse inventory and ensure timely product delivery to customers
- Analyze sales data to understand daily, monthly, and quarterly revenue patterns
- Predict sales and product patterns to increase store profit and inform company goals

Business Rules

- Each customer must be associated with at least one transaction, *as identified by an order number*
- A customer can place multiple orders
- Each transaction will have a unique order number
- There is one, unique contact for each warehouse, supplier, and customer
- An order can include multiple products
- Payment must be when placing an order (i.e., the order and payment day are the same)
- Each order only ships to one customer
- A warehouse stores multiple products
- Every order will be shipped out together (orders will NOT be split for shipment)
- Each supplier only supplies one product
- An order is refunded the same day it is returned (i.e., the return date is the same as the refund date)

Design Requirements

- Create a 3NF-normalized relational database
- Use Crow's Foot notation to show the relationship between entities (e.g., cardinality and participation)
- Implement the database into a logical model using the class SQL server.
- Specify the primary and foreign keys in the diagram by using "PK" and "FK," respectively

Design Decisions

Entity Name	**Why is the Entity Included**	**What is included in this Entity**	**How Entity is Related to Other Entities**
Customer	This database exists to facilitate the connection between customers and suppliers in an online retail shop. The Customer entity is fundamental to the system as it represents the users who purchase goods or services. It allows the platform to store and manage individual customer details, enabling personalized services, order tracking, and targeted marketing. The Customer entity maintains customer profiles and establishes a connection to their orders and communication preferences.	customer_id, location_id contact_id, first_name, last_name	The **Customer** entity is related to the **Contact** entity through the **contact_id** attribute, allowing access to detailed contact information. It is also directly linked to the **Order_Header** entity via the **customer_id** to record which customer placed a particular order. This relationship enables the connection between customer information and their orders.
Order_Header	The Order_Header entity manages the customer's order-level transactions. It captures the overall information about the sale, allowing for sales reporting, billing, and inventory adjustments.	order_id, order_detail_id, order_date, customer_id, payment_id, shipment_id, review_id, return_id	The **Order_Header** entity connects to several entities: the **Customer** entity via **customer_id** to record who placed the order; the **Shipments** entity to track shipment(s) associated with the order; and also the **Order_Detail** entity, **Payment** entity, **Review** entity, and **Return** entity. These relationships ensure that the various details of each order can be traced back to its customer.
Order_Detail	The Order_Detail entity contains the specific itemized order details such as product_id and product_quantity_sold	order_detail_id, product_id, product_quantity_sold	The **Order_Detail** entity links the **Order_Header** table to the **Product** table to track details about the products contained in the order.
Shipments	The Shipments entity ensures efficient tracking	shipment_id, shipment_date,	The **Shipments** entity is linked to the **Order** entity, ensuring

	of the delivery process, which is a key component of online retail operations. This entity allows the system to record shipping details, such as the tracking number, delivery status, and expected delivery date. This information is important to be able to provide customers with real-time updates about their orders and for handling any delivery issues or delays.	carrier, tracking_number, shipment_status, delivery_date, ship_time	that the delivery process is accurately tracked for each specific order.
Contact	The Contact entity contains all communication details for customers, such as phone numbers, email addresses, and shipping locations. This entity will enable efficient customer communication preferences and ensure that the online shop can send notifications, promotional offers, and order updates to the correct and relevant contacts.	contact_id, phone, email	The **Contact** entity is related to the **Customer** entity via the **contact_id** attribute, linking each customer to their preferred contact information. It ensures that communications with customers are conveyed accurately.
Product	The Product entity is a fundamental component of the database that maintains this information, and ensures smooth order processing. Additionally, the inclusion of the Product entity enables tracking of product availability, pricing updates, and sales trends, which are crucial for both suppliers and customers.	product_id, supplier_id, category_id, inventory_id, warehouse_id, product_name, product_price	The **Product** entity is linked to both the **Order_Detail** and **Product_Category** entities, storing information about purchased products. It is also related to the **Supplier** and **Warehouse** entities, which manage supplier relationships, as well as to **Inventory**, which tracks products' inventory status.
Supplier	The Supplier entity is	supplier_id	The **Supplier** entity connects to

	included in the database to represent the vendors who provide products for the platform. Suppliers play a critical role in ensuring that the platform can source and stock products efficiently. By maintaining structured information on suppliers, the system can track sourcing details, manage supplier performance, and streamline the procurement process.	contact_id, supplier_name	the **Product** entity through supplier_id, ensuring that the system accurately tracks which suppliers provide specific products. This supports efficient procurement and supply chain management. It is also connected to **Contact** via the contact_id, ensuring that each supplier's contact information can be managed and accessed.
Inventory	The Inventory entity tracks the stock quantity of each product available in the warehouse. The inventory team may be interested in capturing information on the stock quantity to order optimal quantities into the warehouse. Additionally, customers may be interested in the stock remaining for each product to influence them to purchase products before they run out of stock.	inventory_id, stock_quantity	The **Inventory** entity is directly related to the **Product** entity via the inventory_id attribute, linking each product_id to its stock quantities. It ensures that information regarding product stock is conveyed to customers and the business team accurately.
Warehouse	The team is interested in tracking the name of the warehouse, the address of where each warehouse is located, and the contact information of the warehouse in each location. This entity will enable efficient communication with the correct warehouse regarding products stored and the relevant contact point within the warehouse.	warehouse_id, location_id, contact_id, warehouse_name	The **Warehouse** entity is directly related to the **Location** and **Product** entities so that the business team is able to obtain information about both the location(s) of relevant warehouses and the products that are located there.

Payment	Another key function of the database is to track the transaction status, payment method, and payment amount completed by customers to ensure that every product ordered is paid for. This also helps the team to calculate revenue based on product sales.	payment_id, payment_method, payment_amount, payment_status	The **Payment** entity is related to the **Order_Header** entity in order to tie the payment information and status to each order made by customers. As the customer information may be acquired through the order header entity, the payment entity does not contain this attribute.
Return	Returns are an essential part of retail operations. The Return entity helps track return history, analyze trends (e.g., frequent return reasons), and manage reverse logistics efficiently.	return_id, return_date, return_status, return_reason refund_amount	The **Return** entity connects with **Order_Header** and **Date**, enabling the tracking of customer-initiated returns, linking them to specific orders and products, and managing associated refunds.
Review	The Review entity provides data for product performance analysis and customer engagement. It collects customer feedback to understand sentiment about products and improve future offerings.	review_id, review_date, rating, review_text	The **Review** entity links **Order_Header** and **Date**, allowing customers to provide feedback on purchased products and indirectly verifying purchase details through orders.
Date	The Date entity provides a centralized reference for all dates used in the database (e.g., order dates, shipping dates, return dates). It can support time-based analysis, such as sales trends, shipping delays, or return patterns.	date_id, full_date, day_of_week, date_number, month_name, month_number, quarter_number, calendar_year, fiscal_period	The **Date** entity centralizes date information for key activities, such as order placements, shipments, payments, and returns, ensuring consistency across **Order**, **Shipment**, **Payment**, **Return**, and **Review** entities.
Location	The Location entity centralizes all location data, reducing redundancy and simplifying updates. It can support geographic	location_id, street_address, city, state_province, postal_code, country	The **Location** entity stores addresses for **Customer** and **Warehouse** management.

	analysis (e.g., sales by region, and warehouse locations).		
Product_Category	The Product_Category entity organizes products into logical groups for easier management and customer browsing. It can facilitate sales analysis by category (e.g., dresses vs. accessories).	category_id, product_category	The **Product_Category** entity organizes products into groups, facilitating the classification and analysis of items in the **Product** entity via the category_id attribute.