Shein E-commerce Platform Database

Final Database Specification: Purpose, Business Problems Addressed and Business Rules

Zhiqing Wang, Wenwei Li, Chengran He, Jiayue Han, Xinan Wang

Database Purpose

Shein E-commerce Platform Database aims to manage the sales data produced by Shein and generate business analysis and insights for supply chain department, marketing department, and sales department.

Specifically, the database provides data to calculate the inventory statistics of the Shein e-commerce system. In addition, it contains data to analyze consumers' consumption habits according to customers’ ordering behavior.

Business Problems Addressed

- Support customer profile analysis to improve customer stickiness, attract new customers, and optimize customer conversion rate.
- Provide customer consumption habits analysis to increase sales and benefits with product recommendations.
- Allow inventory department to maintain the supply of products

Business Rules

- Each customer has a unique customer profile.
- Each customer has access to details of his/her orders.
- Each customer can modify his/her payment method.
- Each customer has access to his/her membership.
- Each customer can create his/her wishlist.
- Each customer can keep track of his/her shipment information.
- Each customer may have one customer default payment
- Each customer may have zero or more membership
- Each customer may have zero or more orders
- Each customer will have one or more wishlist
- Each customer will have one or more shopping carts
- Each customer may have zero or more referral histories
- Each order will have exactly one shipment information
- Each order will have exactly one payment information
- Each order may have one returned order
- Each returned order will have one return ship information
- An order can be returned within 30 days of the delivery date (Updated)
- An order can be returned only once (Updated)
- Each order will include one or more products
- Each product will be ordered in one or more orders
- Each product may correspond to zero to one sales product
- Each sales product will correspond to one or more products
- Each product will be included in one or more categories

Design Requirement and response (Created to Professor Simon Wang)

1) **Make sure a relationship has the correct type.**

In the final ERD, we change the relationship from identifying to non-identifying as below.

a. CustomerProfile and MemberShip
b. CustomerProfile and ReferralHistory
c. CustomerProfile and DefaultPayment
d. CustomerProfile and WishList
e. CustomerProfile and ShoppingCart
f. CustomerProfile and Order
g. Order and ShipmentInfo
h. Order and PaymentInfo
i. Order and ReturnedOrder
j. ProductDetail and SalesProduct
k. ProductDetail and ProductCategory
l. ReturnOrder and ReturnShipInfo (Assume business rule to accept ship return only, we will update it in the design document)

In terms of identifying relationships, we avoid many-to-many relationships by creating 4 weak entities, OrderProduct, ReturnProduct, WishListProduct, and ShoppingCartProduct, which satisfies the ERD in the 3rd Normalization Form. Thus, we can track relationships as below.

a. Track the Product quantity in each order, and track the product detail through connecting Order, OrderProduct, and ProductDetail.
b. Track the returned product in each order via ReturnOrder, ReturnProduct, and ProductDetail.
c. Track the product variance in different Wishlist created by customers through WishList, WishListProduct, ProductDetail, and CustomerProfile
d. Track the product change in shopping cart by connecting ShoppingCart, Product, and ShoppinCartProduct

2) **In Entity CustomerProfile, the attribute PaymentSaved seems a composite attribute. Is it about the payment method or payment transactions?**

PaymentSaved refers to information about whether a customer has saved their payment or not, so it is a Boolean data type instead of a composite attribute.

3) **The names in Membership are superfluous and can be removed. They are already in CustomerProfile.**

We have deleted the name in Membership, so we can track membership name through CustomerProfile entity.

4) **Entity WishList should have a one-to-many relationship with Product, with many on Product.**

We assume that a product can appear in different WishLists and a WishList can contain multiple products based on our research on Shein. Thus, the relationship should be many-to-many.

As we described in comment 1, we create a weak entity named WishListProduct to connect the product, customer, and wishlist.

5) Entity ShoppingCart should have a one-to-many relationship with Product, with many on Product.

We assume that a product can appear in different ShoppingCart and a ShoppingCart can contain multiple products based on our research on Shein. Thus, the relationship should be many-to-many.

As we described in comment 1, we create a weak entity named ShoppingCartProduct to connect the product, customer, and shopping cart.

6) **The relationship between CustomerProfile and WishList should have the optional participation on the WishList side.**

We correct the relationship between CustomerProfile and WishList to optional participation on the WishList Side.

7) **Returned Order should be connected to OrderProduct since a customer may only return part of the order.**

Firstly, we change the Product entity to ProductDetail entity. A return request may contain part of the product in the order, and a product can be returned multiple times, so the relationship is many-to-many. As we described in comment 1, we track the returned product with specific OrderID and ReturnID by ReturnOrder weak entity.

8) It is best to store the date of birth in CustomerProfile and calculate the age from it.

We change the Age to BirthDate to analyze the age of the customers.

9) Address in Order is composite and needs to be split into atomic attributes.

Address can be split into street number, city, states, postcode, and country to ensure it is an atomic attribute.

10) Other changes we made to the final ERD

Besides the comments, we made some changes to the ERD, which will be demonstrated in the final design document in the presentation.

a. Add detailed information to track the order shipment, including delivery address and name with atomic attributes
b. Change the payment business rule to an order that can be paid multiple times since some payments may fail. Thus, we add a Boolean Success column to mark the successful transaction
c. Change the PK of the ReferralHistory entity from CustomerID to the ReferralID since a customer may refer to different customers.
d. Delete the name information from the DefaultPayment entity since it duplicates in CustomerProfile
e. Delete the SalesID in ProductDetail to avoid many to many relationship between ProductDetail and SalesProduct.

Design Decisions

No.	Entity Name	Why Entity Included	How Entity is Related to Other Entities
1	CustomerProfile	One of the main purposes of the database is to collect sales data corresponding to individual customers. Important data such as CustomerID, customer first and last name, login email, encrypted password, etc., should be included to identify each customer.	This entity is the core entity in this database. The primary key of CustomerProfile, CustomerID, is related to membership, customer default payment, order, shopping cart, Wishlist, referral history so that we can easily track all the business information about the customers.
2	MemberShip	The team is interested in keeping track of the membership status of each customer, as it could generate business insights	The MemberShip entity is related to the CustomerProfile entity to express the membership information about the customers. The relationship between them is one to many because one

		about differences in purchase patterns between membership/non-membership users.	customer could have different types of memberships. Because it’s also possible that one customer doesn’t have any membership, the membership entity is optional to the CustomerProfile entity. The primary key of MemberShip is MembershipID.
3	CustomerDefaultPayment	Store customer’s preferred payment method for convenience and save time for making a payment.	The CustomerDefaultPayment entity is directly related to the CustomerProfile entity. Because one customer must have exactly one default payment, the relationship between these two entities is one to one, mandatory. The primary key of CustomerDefaultPayment entity is CustomerID
4	SheinOrder	The team is interested in tracking a customer’s purchase frequency and frequently bought product, as these provide insights into consumer behavior.	The order entity is another core entity in this database. It is related to the CustomerProfile entity to describe the customer’s order information. It is also related to the PaymentInfo, ShipmentInfo, OrderProduct and ReturnedOrder entity to let us easily track the information from different aspects about an order. The primary key of Order entity is OrderID
5	ShoppingCart	Count the quantity and price range of the products in the customer's shopping cart, check the price range of the products with the highest ordering frequency of the customer, and recommend products to the customer more pertinently	• The ShoppingCart entity is related to the CustomerProfile entity to express the shopping cart information about the customer. The relationship between them is one too many because one customer could have many carts. The ShoppingCart entity is mandatory to the CustomerProfile entity because every customer must have at least one shopping cart. • It is also related to CartProduct Entity. The primary key of ShoppingCart is CartID.
6	ShipmentInfo	Keep track of shipping information for each order.	The ShipmentInfo entity is directly related to the Order entity. Because one order must have exactly one shipment

			information, the relationship between these two entities is one to one, mandatory. The primary key of ShipmentInfo entity is ShipmentID
7	ReturnOrder	Keep track of a returned order and determine whether a refund can be issued.	• The ReturnedOrder entity is directly related to the Order entity. Because one customer may have one return order to return the stuff they don’t like, the relationship between these two entities is one to one optional. The primary key of ReturnedOrder entity is ReturnID. • A return request may contain part of the product in the order, and a product can be returned multiple times, so the relationship is many-to-many. As we described in comment 1, we track the returned product with specific OrderID and ReturnID by ReturnOrder weak entity.
8	ReturnShipInfo	Keep track of a returned item shipment information, pay attention to delivery information.	The ReturnShipInfo entity is directly related to the ReturnedOrder entity. It supplements the shipping info about the return order. Because one return order must have exactly one ship information, the relationship between these two entities is one to one, mandatory. The primary key of ReturnShipInfo entity is ShipmentID
9	ProductDetail	Categorize the name and quantity of products in the product library, easy to find and count inventory and sales	The product entity is related to order product, salesproduct and category. It contains essential information for a product. The primary key for this entity is ProductID.
10	PaymentInfo	Store customers' payment methods, focusing on checking whether there are multiple accounts registered at the same	The PaymentInfo entity is related to the Order entity. According to the business rule, an order made by a customer will have a payment, which may be different from the default customer payment

		address, and merging accounts if there are.	method. Thus, the relationship from Order to PaymentInfo is one and only one, and the PaymentInfo to Order is also one and only one relationship.
11	SalesProduct	The team is interested in tracking the impact of promotions on product sales.	The SalesProduct entity is related to Product entity. A sales product must be included in the product category, the product has unique salesID related to the product ID, so these two have one-to-many relationship.
12	WishList	Allow consumers to easily categorize the things they desire, eliminating the need to utilize the shopping cart for rapid recall.	• The Wishlist entity is directly related to CustomerID. The primary key is WishlistID.. A customer can create a different Wishlist at a different date, so the relationship from Customer to WishList is one and more, while the WishList to Customer is one and only one. • There is a weak entity named WishListProduct to connect the product, customer, and wishlist.
13	Category	Catalog is a standardized model that integrates various vendor-specific products in a domain.	The Category entity is directly related to Product entity. According to the business rule, each product will be included in one or more categories, so Product entity and Category entity have one-to-many relationships. The primary key is CategoryID.
14	ReferralHistory	Encourage customers to recommend Shein APP to friends through cash rewards or discounts, increase the number of Shein customers, and track the referral history.	The referral history is related to customerID, a current customer can refer to their friends and this can happen repeatedly. The referral history entity and the customerID entity have a one-to-many relationship.
15	OrderProduct	Because there is a many-to-many relationship between order entity and product entity, we create this associative entity to describe each entity	The OrderProduct entity is related to the Order entity and Product entity. The Order entity and OrderProduct entity has one-to-many relationship. Also, the Product entity and OrderProduct entity have one-to- many relationship. The primary key of

			OrderProduct entity is OrderID and ProductID.
16	ReturnProduct	A weak entity to associate entity between returned order and product.	It is connected to ReturnOrder and ProductDetail to show that some products in an order can be returned due to some reasons.
17	WishListProduct	A weak entity to associate entity between returned order and product.	It is connected to WishList and ProductDetail to show that some products can be shown in many WishList produced by customers.
18	ShoppingCartProduct	A weak entity to associate entity between returned order and product.	It is connected to ShoppingCart and ProductDetail to show that a product can be shown in many shopping carts created by customers.