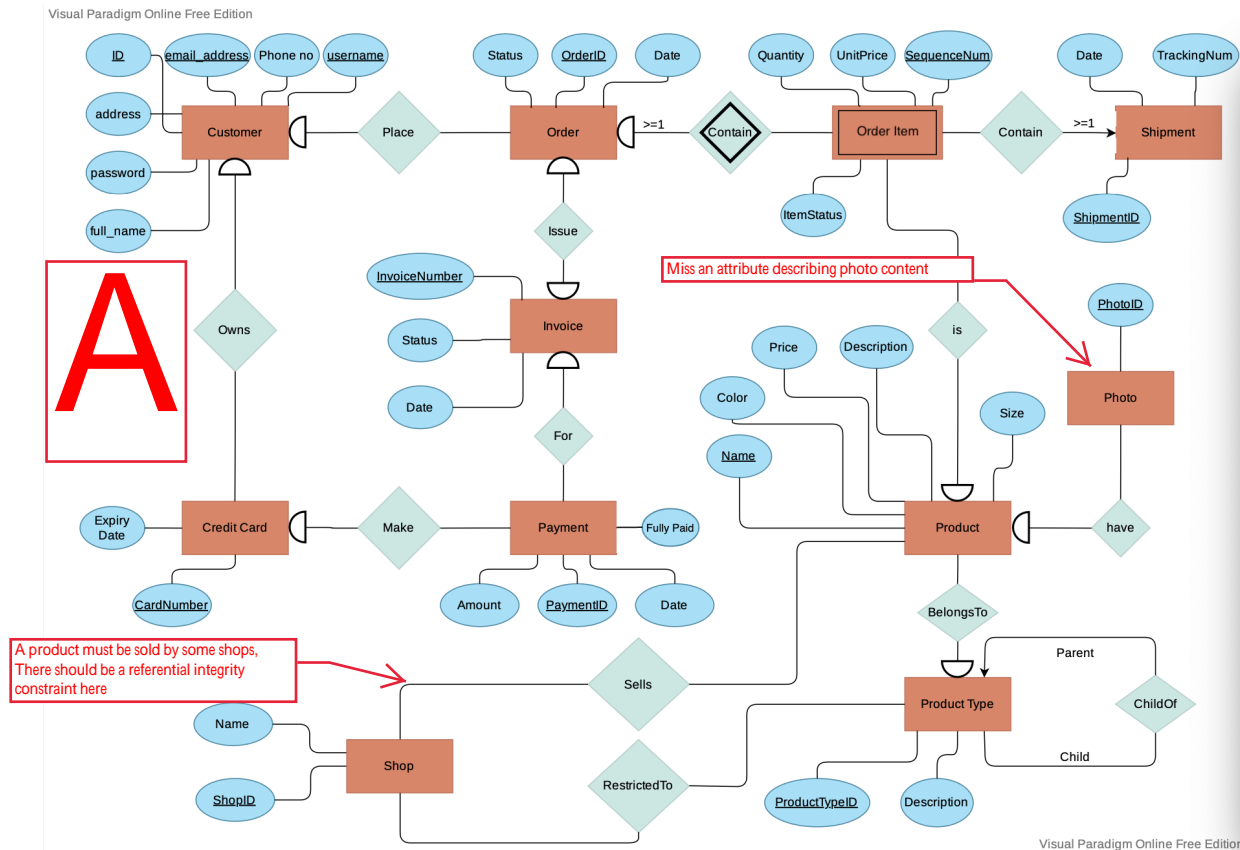


SSP4 - GROUP 1

CZ2007 Database Lab 1

ER Diagram



Design Rationale

The ER diagram can be divided into 3 logical sections (Customer-Payment, Shop-Product, Orders-Shipment) that interact with each other.

Customer-Payment

This section deals with the relationship between customers and payment methods. It covers how a customer can use credit card(s) to pay for an order.

Entities and Relationships

Customer – This entity stores information about the customer. Username, id or email address can uniquely identify a customer.

Credit Card – This entity stores information about credit cards. The CardNumber can uniquely identify a credit card.

Each customer can own multiple credit cards, however each credit card can only be owned by one and only one customer. Therefore, the Credit Card-Customer relationship is a many-to-one relationship.

Payment – This entity stores information of any payment made by a customer for an invoice. Each payment has a PaymentID that uniquely identifies a payment.

Invoice – This entity stores information of any invoice. One invoice is issued for every order. Each invoice is identified by an Invoice number.

Customers can use a credit card to make payment(s) for an invoice. Each payment must only be made by one and only one credit card. Each credit card can make multiple payment. Payment-CreditCard is a many-to-one relationship.

Each invoice can have multiple payments, however we assume that each payment must only be for one and only one invoice. Therefore, Payment-Invoice relationship is a many-to-one relationship.

Orders-Shipment

This section covers the structure of an order.

Entities and Relationships

Order – This entity stores the information of orders placed by customers. Orders are uniquely identified by its OrderID.

OrderItem – This entity stores information of each order item in an order. Order items within an order are uniquely identified by its SequenceNum. This is a weak entity set that is associated with Order.

Each Order can have one or more OrderItems. Each OrderItem must be tagged to an order. OrderItem-Order relationship is a many to one relationship.

Shipment – this entity stores the information of a shipment. ShipmentID uniquely identifies a shipment.

Each shipment can contain multiple OrderItems. We assume each OrderItems will only be shipped in one shipment. For example, if an OrderItem contains 5 quantity of a particular product, all 5 item must be shipped in a single shipment. OrderItem-Shipment is a many-to-one relationship

Shop-Product

This section covers the structure of a shop and its product.

Entities and Relationships

Shop – this entity stores information of a shop in the platform. Each shop is uniquely identified by its ShopID

Product – this entity stores information of a product. Each product is identified by its Name.

Each shop can sell multiple different products. We assume each product can be sold on multiple different shops. Shop-Product is a many-many relationship.

ProductType – This entity stores information of a product type. ProductTypeID uniquely identifies a product type.

Each product must belong to one and only ProductType group. Each ProductType group can contain multiple products. Product-ProductType is a many-to-one relationship.

Each ProductType can be a child of at most one ProductType. A parent ProductType can have many children ProductType. Therefore, ProductType_Child-ProductType_Parent is a many-to-one relationship. We assume that there is only 1 ProductType at the top of the hierarchy and all other ProductType falls below that ProductType.

Shops are restricted to selling some ProductType groups. Each ProductType can be sold on different Shop. Shops-ProductType is a many-to-many relationship. When a shop is restricted to selling a certain ProductType, the shop is able to sell the ProductType and all child under the ProductType group. In the case where the shop has no restrictions to selling any product, then it is restricted to selling the ProductType group at the top of the hierarchy.

Photo – this entity stores information on the photo associated with the product. PhotoID uniquely identifies a photo.

A Product can have multiple photos associated with it. We assume no two products will use the same photo and each photo will only be associated with one and only 1 product. Product-Photo is a many-to-many relationship.

Relationship between the 3 Logical Sections

Each Customer can place orders. An order must be placed by one and only one customer. Customer-Order is a many-to-one relationship.

Every Order issues one and only one Invoice. Each Invoice is issued for one and only one Order. Invoice-Order is a one-to-one relationship.

Each OrderItem contains a Product. Each Product can be sold in different OrderItem. OrderItem-Product is a many-to-one relationship.

Appendix D

Individual Contribution Form

Name	Individual Contribution for Submission 1 (Lab 1)	Percentage of Contribution (100% in total)
Ang Guang Yao	Participate in discussion of ER diagram, wrote description for ER diagram	20
Reeves Chiu	Participate in discussion of ER diagram, and facilitate the team	20
Chai Wen Xuan	Participate in discussion of ER diagram, do checking for the ER diagram.	20
Ingale Omkar	Participate in discussion of ER diagram, created the diagram using online/locally downloaded editing tools	20
Ivan Pua	Participate in discussion of ER diagram, contribute ideas to improve ER diagram	20