Design and Structure of Amazon's E-commerce Database

1st Juan David Quiroga
Universidad Distrital Francisco Jose de Caldas, Bogotá, Colombia
2nd Luis Alejandro Morales
Universidad Distrital Francisco Jose de Caldas, Bogotá, Colombia

Abstract

This document provides a concise overview of how one of the most recognized Amazon web pages is constructed using the entity-relationship model (DER). It highlights components such as purchasing, sales, suppliers, among others, which will be discussed below.

Index Terms—Amazon, database, DER.

0.1 INTRODUCTION

Amazon is one of the world's largest e-commerce platforms, founded in 1994 by Jeff Bezos. Over the years, Amazon has built a reputation based on convenience, speed of delivery, and a wide variety of products, positioning it as a global leader in online commerce. Amazon's success is largely attributed to its ability to handle millions of daily transactions efficiently, which necessitates a robust and scalable database system. This system enables the storage, organization, and access to large volumes of product, customer, and transaction information, ensuring data is processed quickly and accurately. This paper aims to describe Amazon's database design in the context of its e-commerce operation, focusing on the main entities and relationships that facilitate the management of vast amounts of data handled by the platform.

0.2 METHOD AND MATERIALS

0.2.1 Taking a Look at the DER

The method used to describe and model the Amazon database is the entity-relationship diagram (DER), a widely recognized technique in software and database engineering. The DER graphically represents the logical structure of a database, showing the entities that compose it, the attributes of those entities, and the relationships between them.

In an entity-relationship diagram, the entities represent the main objects to be stored in the database, such as "Customers", "Products", and "Orders". Each entity contains specific attributes that describe its properties, such as name, address, or order date. The relationships between entities define how they interact within the system, for example, a customer may place multiple orders, or an order may be associated with multiple products.

Using a DER to model Amazon's database simplifies understanding its internal structure, ensuring that data is organized efficiently and that the relationships between entities accurately reflect real e-commerce operations. This is critical for optimizing database queries and overall system performance.

1 DEFINE COMPONENTS

The basis for this is born between:

- **Users**
  - Customers: Interact with the site to search, evaluate, and purchase products. They can leave reviews and ratings.
  - Sellers: Offer products on the platform.
- **Shopping Cart System**
  - User Interaction: Allows users to add products and view their selection before proceeding to checkout.
- **Payment System**

  - Payment Method Interaction: Communicates with various payment gateways (such as credit cards, PayPal) to process transactions.

- **Order Management**
  - Status Update: Allows users to track the status of their orders (processing, shipped, delivered).

2 DEFINE ENTITIES

The following entities are defined:

- Customer (Usuario)
- Product (Producto)
- Category (Categoría de Producto)
- Order (Pedido)
- ShoppingCart (Carrito de Compras)
- PaymentMethod (Método de Pago)
- Review (Reseña)
- Shipping (Envío)
- Offer (Oferta)
- Seller (Vendedor)
- SearchHistory (Historial de Búsquedas)
- ProductRecommendations (Recomendaciones de Productos)
- Returns (Devoluciones)
- Coupons (Cupones)
- Order-Items (Organizazdor-items)

3 DEFINE ATTRIBUTES FOR EACH ENTITY

Each entity is associated with specific attributes that define its characteristics. The key attributes for each entity are as follows:

- **Customer:** User ID (PK), Full Name, Email (UNIQUE), Shipping Address, Phone, Registration Date.
- **Product:** Product ID (PK), Product Name, Description, Price, Quantity Available, Category ID (FK → Category.ID), Seller ID (FK → Seller.ID).
- **Category:** Category ID (PK), Category Name, Description.
- **Order:** Order ID (PK), Order Date, Customer ID (FK → Customer.ID), Total Amount, Order Status, Payment Method ID (FK → PaymentMethod.ID), Shipping ID (FK → Shipping.ID).
- **Shopping Cart:** Cart ID (PK), Customer ID (FK → Customer.ID).
- **Shopping Cart Product:** Cart ID (FK → ShoppingCart.ID), Product ID (FK → Product.ID), Quantity, PK: (Cart ID, Product ID).
- **Payment Method:** Payment Method ID (PK), Payment Type, Customer ID (FK → Customer.ID).
- **Review:** Review ID (PK), Rating, Comment, Review Date, Customer ID (FK → Customer.ID), Product ID (FK → Product.ID).
- **Shipping:** Shipping ID (PK), Shipping Company, Shipping Date, Estimated Delivery Date, Shipping Cost.
- **Offer:** Offer ID (PK), Discount, Start Date, End Date.
- **Seller:** Seller ID (PK), Seller Name, Seller Type, Seller Rating.
- **Search History:** Search ID (PK), Search Term, Search Date, Customer ID (FK → Customer.ID).
- **Product Recommendations:** Recommendation ID (PK), Customer ID (FK → Customer.ID), Recommended Product ID (FK → Product.ID), Recommendation Date.
- **Returns:** Return ID (PK), Return Date, Return Reason, Return Status, Order Item ID (FK → OrderItems.ID).

- **Coupons:** Coupon ID (PK), Discount Code, Discount Value, Expiration Date.
- **Order Items:** Order Item ID (PK), Order ID (FK → Order.ID), Product ID (FK → Product.ID), Quantity, Price at Purchase, Coupon ID (FK → Coupons.ID), Offer ID (FK → Offer.ID).

4 DEFINE RELATIONSHIPS

The following table shows the relationships that the entities have with each other, which we can then see what type they are related to.

	Customer	Product	Category	Order	ShoppingCart	ShoppingCartProduct	PaymentMethod	Review	Shipping	Offer	Seller	SearchHistory	ProductRecommendations	Returns	Coupons	OrderItems
Customer				X	X		X	X				X	X			
Product			X			X		X		X	X		X		X	X
Category		X														
Order	X						X		X							X
ShoppingCart	X					X										
ShoppingCartProduct		X			X											
PaymentMethod	X			X												
Review	X	X														
Shipping				X												
Offer		X														
Seller		X														
SearchHistory	X															
ProductRecommendation	X	X														
Returns																X
Coupons		X														X
OrderItems		X		X										X	X	

Figure 1: Entity Relationship Table

5 DEFINE TYPES OF RELATIONSHIPS

6 FIRST VIEW OF THE DIAGRAM

7 DIVIDE MANY-TO-MANY RELATIONSHIPS

Within our first design (DER), we find that between the entities "Products" and "ShoppingCart" there is a many-to-many relationship since a shopping cart can contain many products and a product can be in many carts. We must eliminate this relationship since it can generate problems in the queries. That is why we implemented a new entity called "ShoppingCart_Product" with the following attributes:

Entidad 1	Entidad 2	Tipo de Relación	Descripción
Customer	Order	1:N	Un cliente puede hacer múltiples pedidos, pero un pedido pertenece a un solo cliente.
Customer	Shopping Cart	1:1	Cada cliente tiene un único carrito de compras.
Customer	Payment Method	1:N	Un cliente puede tener múltiples métodos de pago, pero un método de pago pertenece a un solo cliente.
Customer	Review	1:N	Un cliente puede escribir múltiples reseñas, pero cada reseña pertenece a un solo cliente.
Customer	Search History	1:N	Un cliente puede tener múltiples búsquedas registradas, pero cada búsqueda pertenece a un solo cliente.
Customer	Product Recommendations	1:N	Un cliente puede recibir múltiples recomendaciones de productos, pero cada recomendación pertenece a un solo cliente.
Product	Category	N:1	Un producto pertenece a una sola categoría, pero una categoría puede tener múltiples productos.
Product	Review	1:N	Un producto puede tener múltiples reseñas, pero cada reseña pertenece a un solo producto.
Product	Shopping Cart Product	N:M	Un producto puede estar en múltiples carritos y un carrito puede contener múltiples productos.
Product	Order Items	N:M	Un producto puede estar en múltiples pedidos y un pedido puede contener múltiples productos.
Product	Coupons	N:1	Un cupón puede aplicarse a un solo producto, pero un producto puede tener múltiples cupones disponibles.
Category	Product	1:N	Una categoría puede tener múltiples productos, pero cada producto pertenece a una sola categoría.
Order	Customer	N:1	Un pedido pertenece a un solo cliente, pero un cliente puede hacer múltiples pedidos.
Order	Payment Method	N:1	Un pedido utiliza un solo método de pago, pero un método de pago puede usarse en múltiples pedidos.
Order	Shipping	1:1	Un pedido tiene un único envío asociado.
Order	Order Items	1:N	Un pedido puede contener múltiples productos.
Shopping Cart	Customer	1:1	Cada cliente tiene un único carrito de compras.
Shopping Cart	Shopping Cart Product	1:N	Un carrito de compras puede contener múltiples productos.
Shopping Cart Product	Product	N:M	Un producto puede estar en múltiples carritos, y un carrito puede contener múltiples productos.
Payment Method	Customer	N:1	Un cliente puede tener múltiples métodos de pago.
Payment Method	Order	1:N	Un pedido utiliza un solo método de pago, pero un método de pago puede usarse en múltiples pedidos.
Review	Customer	N:1	Una reseña pertenece a un solo cliente, pero un cliente puede hacer múltiples reseñas.
Review	Product	N:1	Una reseña pertenece a un solo producto, pero un producto puede tener múltiples reseñas.
Shipping	Order	1:1	Cada pedido tiene un único envío.
Offer	Product	1:N	Un producto puede tener múltiples ofertas, pero cada oferta está asociada a un solo producto.
Seller	Product	1:N	Un vendedor puede vender múltiples productos, pero un producto pertenece a un solo vendedor.
Search History	Customer	N:1	Una búsqueda pertenece a un solo cliente, pero un cliente puede tener múltiples búsquedas registradas.
Product Recommendations	Customer	N:1	Una recomendación pertenece a un solo cliente, pero un cliente puede recibir múltiples recomendaciones.
Returns	Order Items	1:1	Cada devolución está asociada a un único producto dentro de un pedido.
Coupons	Order Items	N:1	Un cupón puede aplicarse a un solo producto dentro de un pedido.
Order Items	Order	N:1	Un pedido puede contener múltiples productos, pero cada producto dentro del pedido pertenece a un solo pedido.
Order Items	Product	N:1	Un producto dentro de un pedido pertenece a un solo producto, pero un producto puede estar en múltiples pedidos.

Figure 2: Types of Relationships

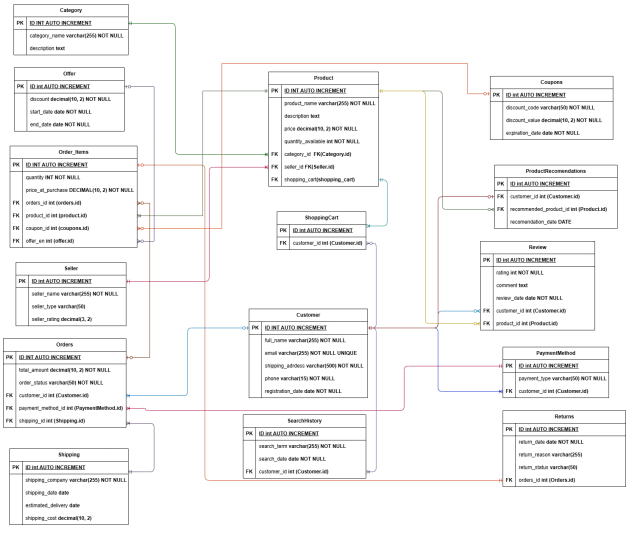

Figure 3: First View of the Entity-Relationship Diagram

- **cart_id**: A foreign key referencing the ShoppingCart entity to identify which shopping cart the product belongs to.
- **product_id**: A foreign key referencing the Product entity to identify which product is being added to the shopping cart.
- **quantity**: An integer attribute indicating the number of units of the product that the customer wants to purchase.

In this way, the relationships will be one-to-many.

8 SECOND VIEW OF THE DIAGRAM

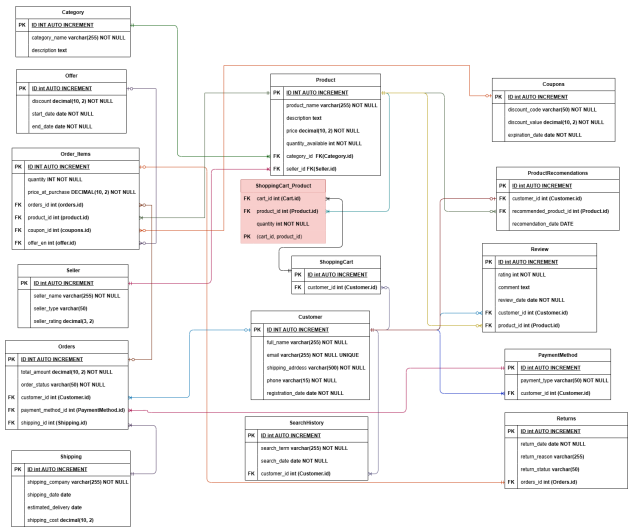

Figure 4: Second View of the Entity-Relationship Diagram

9 OBTAIN THE DATA STRUCTURE

- **Customer** (Cliente):
  - id (int, PK), full_name (varchar), email (varchar, UNIQUE), shipping_address (varchar), phone (varchar), registration_date (date)

- **Product** (Producto):
  - id (int, PK), product_name (varchar), description (text), price (decimal), quantity_available (int), category_id (int, FK → Category.id), seller_id (int, FK → Seller.id)
- **Category** (Categoría):
  - id (int, PK), category_name (varchar), description (text)
- **Order** (Pedido):
  - id (int, PK), total_amount (decimal), order_status (varchar), customer_id (int, FK → Customer.id), payment_method_id (int, FK → PaymentMethod.id), shipping_id (int, FK → Shipping.id)
- **ShoppingCart** (Carrito de Compras):
  - id (int, PK), customer_id (int, FK → Customer.id)
- **ShoppingCart_Product**:
  - cart_id (int, FK → ShoppingCart.id), product_id (int, FK → Product.id), quantity (int), PK: (cart_id, product_id)
- **PaymentMethod** (Método de Pago):
  - id (int, PK), payment_type (varchar), customer_id (int, FK → Customer.id)
- **Review** (Opinión):
  - id (int, PK), rating (int), comment (text), review_date (date), customer_id (int, FK → Customer.id), product_id (int, FK → Product.id)
- **Shipping** (Envío):
  - id (int, PK), shipping_company (varchar), shipping_date (date), estimated_delivery (date), shipping_cost (decimal)
- **Offer** (Oferta):
  - id (int, PK), discount (decimal), start_date (date), end_date (date)
- **Seller** (Vendedor):
  - id (int, PK), seller_name (varchar), seller_type (varchar), seller_rating (decimal)

- **SearchHistory** (Historial de Búsquedas):
  - id (int, PK), search_term (varchar), search_date (date), customer_id (int, FK → Customer.id)
- **ProductRecommendations** (Recomendaciones de Productos):
  - id (int, PK), customer_id (int, FK → Customer.id), recommended_product_id (int, FK → Product.id), recommendation_date (date)
- **Returns** (Devoluciones):
  - id (int, PK), return_date (date), return_reason (varchar), return_status (varchar), order_item_id (int, FK → Order_Items.id)
- **Coupons** (Cupones):
  - id (int, PK), discount_code (varchar), discount_value (decimal), expiration_date (date)
- **Order_Items** (Detalles del Pedido):
  - id (int, PK), order_id (int, FK → Order.id), product_id (int, FK → Product.id), quantity (int), price_at_purchase (decimal), coupon_id (int, FK → Coupons.id), offer_id (int, FK → Offer.id)

10 DEFINE DATA AND COMPONENT PROPERTIES

Finally, each attribute and relationship in the database must be defined with its respective properties, such as constraints (e.g., primary keys, foreign keys) and data types (e.g. inte.g.er, string, date).

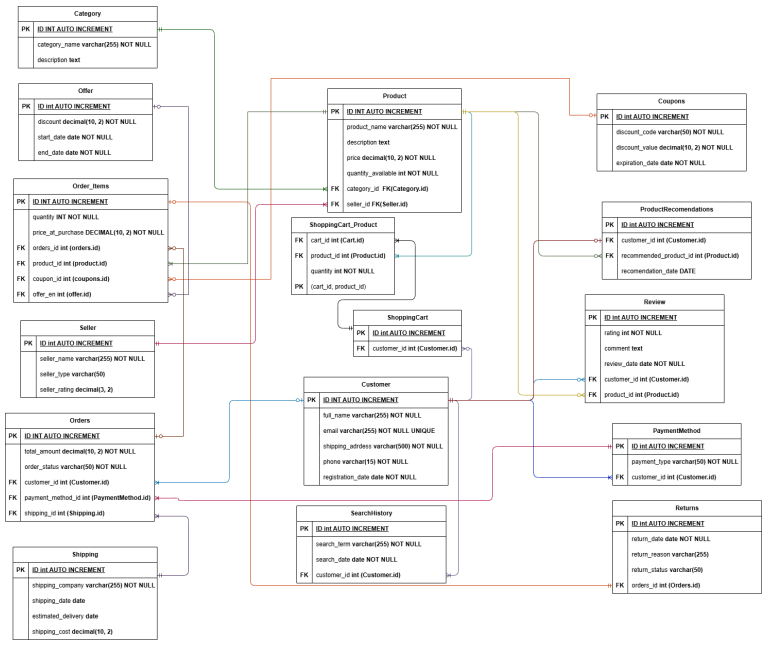

Figure 5: Second View of the Entity-Relationship Diagram

11 CONCLUSIONS

The entity-relationship model is a powerful tool for representing the underlying structure of a database, as shown in Amazon's e-commerce platform. By clearly defining entities, attributes, and relationships, we can create a robust framework that can effectively manage the vast amounts of data generated by the platform. This paper provides a solid foundation for understanding how Amazon's database operates, facilitating further research and development in the field of e-Commerce.

12 REFERENCES

1. Amazon, 2024. *Annual Report*.
2. R. Elmasri and S. Navathe, 2016. *Fundamentals of Database Systems*. 7th ed. Pearson.
3. M. Blaha and J. Premerlani, 2004. *Object-Oriented Modeling and*

Design with UML. 3rd ed. Pearson.

4. J. L. V. P. de Jesus, M. S. V. A. D. de Lima, 2021. "A Review of E-commerce Models: A Case Study of Amazon," *Journal of Retailing and Consumer Services*, vol. 58, p. 102279.