# Design and Structure of Amazon's E-commerce Database $\beta$-$\nu$-$\mu$

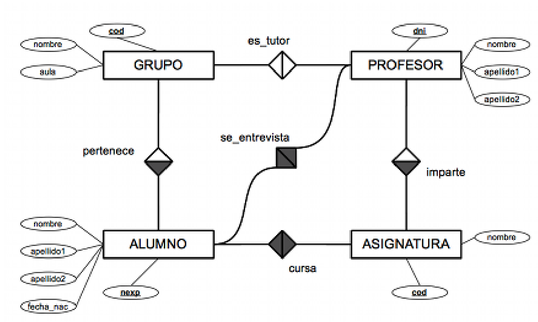
**Juan David Quiroga[1], Luis Alejandro Morales[2]**
Universidad Distrital Francisco Jose de Caldas
jdquirogag@udistrital.edu.co[1], lamoralesn@udistrital.edu.co[2]

## Introduction

The e-commerce landscape has experienced unprecedented growth in recent years, with Amazon emerging as a leader in transforming the online shopping experience. As one of the largest global retailers, Amazon manages an enormous volume of transactions, product information, and user data daily, necessitating a robust and scalable database system capable of efficiently handling this complexity. Traditional relational databases often face challenges such as performance bottlenecks and scalability issues when processing high transaction volumes and extensive product catalogs. Consequently, it is crucial to design a database that not only stores customer and product data but also ensures rapid access and data integrity. This poster explores the challenges inherent in Amazon's database architecture and presents a solution based on an Entity-Relationship Diagram (DER). By clearly modeling the key entities—such as Customers, Products, Orders, and Shopping Carts—and defining their relationships, we aim to create an efficient database structure that enhances query performance and supports the seamless operations of the e-commerce platform.

## Goal

The primary goal of this research is to design an efficient and scalable database structure for Amazon's e-commerce operations. The research question focuses on how to effectively model the relationships between key entities such as customers, products, and orders, while the expected final product is a comprehensive DER that illustrates this structure.




## Proposed Solution

The proposed solution utilizes an Entity-Relationship Diagram (DER) as the foundation for designing Amazon's database. The DER provides a visual representation of the key entities involved in the e-commerce ecosystem and illustrates their attributes and relationships. This structured approach follows a systematic methodology encompassing ten steps, which include:

1. **Define components:** Identifying the main entities such as Customer, Product, Order, etc.
2. **Define entities:** Specifying each entity's role and importance in the database.
3. **Define attributes for each entity:** Listing the necessary attributes for each entity.
4. **Define relationships:** Establishing how these entities interact with each other.
5. **Define types of relationships:** Clarifying the nature of relationships (e.g., one-to-many).
6. **First view of the diagram:** Creating an initial DER to visualize the entities and their relationships.
7. **Divide many-to-many relationships:** Simplifying complex relationships into manageable structures.
8. **Second view of the diagram:** Updating the DER based on optimizations and feedback.
9. **Obtain the data structure:** Determining the tables and columns based on the DER.
10. **Define data and component properties:** Specifying data types and constraints for each component.

## Results

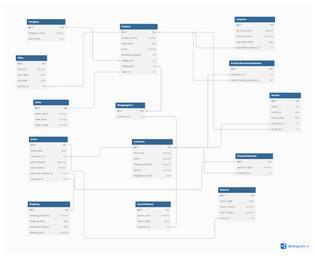
1. Define components
   - **Users:**
     - **Customers:** Interact with the site to search, evaluate, and purchase products; can leave reviews.
     - **Sellers:** Offer products on the platform.
   - **Shopping Cart System:** Enables users to add products and view selections before checkout.
   - **Payment System:** Interfaces with payment gateways (e.g., credit cards, PayPal) for transaction processing.
   - **Order Management:** Allows users to track order status (processing, shipped, delivered).
   - **Recommendation System:** Provides personalized product suggestions based on user behavior.
   - **Review and Rating System:** Facilitates users in leaving comments and ratings on products.
   - **Logistics and Shipping:** Manages inventory and coordinates product shipping.
   - **Promotions:** Implements discounts and coupons to encourage customer loyalty.
2. **Define entities and atributtes:** below are some of the entities used in this diagram
   - 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)
3. **Define relationships:**

| | Customer | Product | Category | Order | ShoppingCart | PaymentMethod | Review | Shipping | Offer | Seller | SearchHistory | ProductRecommendations | Returns | Coupons |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Customer | X | | | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | | |
| Product | | X | ✓ | | | | ✓ | | ✓ | ✓ | | | | ✓ |
| Category | | | X | | | | | | | | | | | |
| Order | ✓ | | | X | | | | ✓ | | | | | ✓ | |
| ShoppingCart | ✓ | | | | X | | | | | | | | | |
| PaymentMethod | ✓ | | | | | X | | | | | | | | |
| Review | ✓ | ✓ | | | | | X | | | | | | | |
| Shipping | | | | ✓ | | | | X | | | | | | |
| Offer | | ✓ | | | | | | | X | | | | | |
| Seller | | ✓ | | | | | | | | X | | | | |
| SearchHistory | ✓ | | | | | | | | | | X | | | |
| ProductRecommendations | ✓ | ✓ | | | | | | | | | | X | | |
| Returns | | | | ✓ | | | | | | | | | X | |
| Coupons | | ✓ | | | | | | | | | | | | X |

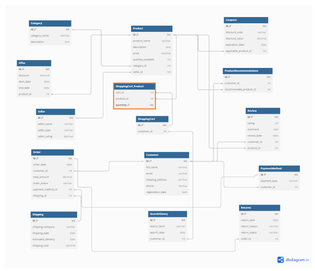
4. **Define types of relationships:**

| Relationship | Type |
|---|---|
| Customer - order | 1 to many |
| Customer - ShoppingCart | 1 to 1 |
| Customer - PaymentMethod | 1 to many |
| Customer - Review | 1 to many |
| Customer - SearchHistory | 1 to many |
| Customer - ProductRecomendations | 1 to many |
| Product - Category | Many to 1 |
| Product - ShopingCart_Product | Many to many |
| Product - Review | 1 to many |
| Product - Ofter | 1 to many |
| Product - Cupons | 1 to many |
| Seller - Product | 1 to many |
| Order - PaymentMethod | 1 to 1 |
| Order - Shipping | 1 to 1 |
| Order - Returns | 1 to many |

5. **First view of the diagram:**



6. **Second view of the diagram with many-to-many relationships divided:**



7. **Define data and component properties and structure:**

| Entity | Attribute | Data Type | Constraints |
|---|---|---|---|
| **Customer** | id | int | PK, Auto Increment |
| | full_name | varchar(255) | NOT NULL |
| | email | varchar(255) | NOT NULL, UNIQUE |
| | shipping_address | varchar(500) | NOT NULL |
| | phone | varchar(15) | NOT NULL |
| | registration_date | date | NOT NULL |
| **Product** | id | int | PK, Auto Increment |
| | product_name | varchar(255) | NOT NULL |
| | description | text | |
| | price | decimal(10, 2) | NOT NULL |
| | quantity_available | int | NOT NULL |
| | category_id | int | FK (Category.id) |
| | seller_id | int | FK (Seller.id) |
| **Category** | id | int | PK, Auto Increment |
| | category_name | varchar(255) | NOT NULL |
| | description | text | |
| **Order** | id | int | PK, Auto Increment |
| | order_date | date | NOT NULL |
| | customer_id | int | FK (Customer.id) |
| | total_amount | decimal(10, 2) | NOT NULL |
| | order_status | varchar(50) | NOT NULL |
| | payment_method_id | int | FK (PayamentMethod.id) |
| | shipping_id | int | FK (Shipping.id) |
| **ShoppingCart** | id | int | PK, Auto Increment |
| | customer_id | int | FK (Customer.id) |
| **ShoppingCart_Product** | cart_id | int | FK (ShoppingCart.id) |
| | product_id | int | FK (Product.id) |
| | quantity | int | |
| **PaymentMethod** | id | int | PK, Auto Increment |
| | payment_type | varchar(50) | NOT NULL |
| | customer_id | int | FK (Customer.id) |
| **Review** | id | int | PK, Auto Increment |
| | rating | int | NOT NULL |
| | comment | text | |
| | review_date | date | NOT NULL |
| | customer_id | int | FK (Customer.id) |
| | product_id | int | FK (Product.id) |
| **Shipping** | id | int | PK, Auto Increment |
| | shipping_company | varchar(255) | NOT NULL |
| | shipping_date | date | |
| | estimated_delivery | date | |
| | shipping_cost | decimal(10, 2) | |
| **Offer** | id | int | PK, Auto Increment |
| | discount | decimal(10, 2) | NOT NULL |
| | start_date | date | NOT NULL |
| | end_date | date | NOT NULL |
| | product_id | int | FK (Product.id) |
| **Seller** | id | int | PK, Auto Increment |
| | seller_name | varchar(255) | NOT NULL |
| | seller_type | varchar(50) | |
| | seller_rating | decimal(3, 2) | |
| **SearchHistory** | id | int | PK, Auto Increment |
| | search_term | varchar(255) | NOT NULL |
| | search_date | date | NOT NULL |
| | customer_id | int | FK (Customer.id) |
| **ProductRecommendations** | id | int | PK, Auto Increment |
| | customer_id | int | FK (Customer.id) |
| | commended_product | int | FK (Product.id) |
| **Returns** | id | int | PK, Auto Increment |
| | return_date | date | NOT NULL |
| | return_reason | varchar(255) | |
| | return_status | varchar(50) | |
| | order_id | int | FK (Order.id) |
| **Coupons** | id | int | PK, Auto Increment |
| | discount_code | varchar(50) | NOT NULL |
| | discount_value | decimal(10, 2) | NOT NULL |
| | expiration_date | date | NOT NULL |
| | applicable_product_id | int | FK (Product.id) |

## Conclusions

The use of entity-relationship diagrams (DERs) is crucial for effective database design, particularly for e-commerce platforms like Amazon. DERs offer a clear visual depiction of the entities within the system, along with their attributes and relationships. This visualization aids in comprehending the database structure and ensures that all components are logically and coherently interconnected.

For Amazon, the complexity of its data ecosystem requires a meticulously planned and structured approach. Proposing a database that incorporates essential entities such as "Customers," "Products," "Orders," and "Shopping Carts" enables efficient management of daily operations and enhances the user experience. By defining relationships, especially between products and shopping carts, common redundancy issues can be addressed, and referential integrity can be preserved.

## Bibliography

Edraw Software. (n.d.). *ER Diagram.* Recuperado de https://www.edrawsoft.com/es/er-diagram/

Universidad de Buenos Aires. (n.d.). *Diagrama Entidad Relación.* Recuperado de https://repositorio.ub.edu.ar/handle/123456789/5155