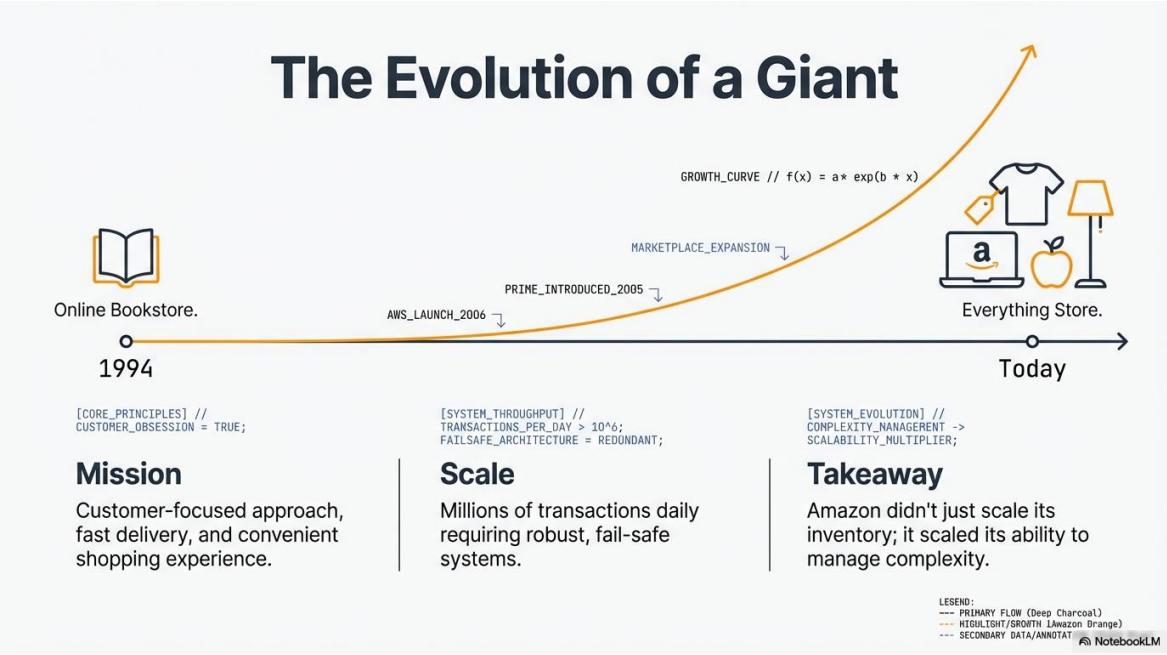
Product Dissection for Amazon

Company Overview:

Amazon was founded by Jeff Bezos in 1994, starting as an online bookstore. Today, it has grown into the world's largest e-commerce platform, offering millions of products across categories like electronics, books, clothing, groceries, and more. Amazon is known for its customer-focused approach, fast delivery, and convenient shopping experience.



Product Dissection and Real-World Problems Solved by Amazon:

Amazon's product discovery feature helps users explore millions of products easily. Users can search by category, filter by price, read customer reviews, and compare products before making a purchase decision.

Amazon offers a wide range of products from multiple sellers, giving users endless choices. The customer review system allows buyers to check product ratings and feedback from other customers, helping them make informed decisions and saving time.

Amazon provides real-time order tracking so users can monitor their package from the warehouse to their doorstep. The platform also offers flexible payment options and easy return policies.

Conclusion: Amazon's product design solves real-life problems by creating a platform where people can find anything they need, shop conveniently from home, and trust the quality through customer reviews.

Case Study: Real-World Problems and Amazon's Innovative Solutions

Problem 1: Limited Product Access

Real-World Challenge: Traditional stores have limited shelf space and can only stock popular items. Customers looking for specific or niche products often struggle to find them locally, especially in smaller cities or rural areas.

Amazon's Solution:

Amazon solved this by creating an online marketplace where millions of products can be listed without physical space limitations. Users can find almost any product they need, from popular items to rare books or specialized equipment. Amazon's powerful search and filter system helps users quickly find exactly what they're looking for among millions of options.

Problem 2: Trust and Quality Concerns

Real-World Challenge: When shopping online, customers cannot physically see or touch products before buying. This creates uncertainty about product quality, whether the item matches the description, and if the seller is reliable.

Amazon's Solution:

Amazon introduced customer reviews and ratings where buyers can share their honest experiences. The 'Verified Purchase' badge shows reviews from actual buyers, making feedback more trustworthy. Amazon also offers an A-to-Z Guarantee that protects

customers if sellers don't deliver as promised, and provides easy return options with prepaid shipping labels. This builds trust and reduces the risk of online shopping.

Problem 3: Delivery Time and Convenience

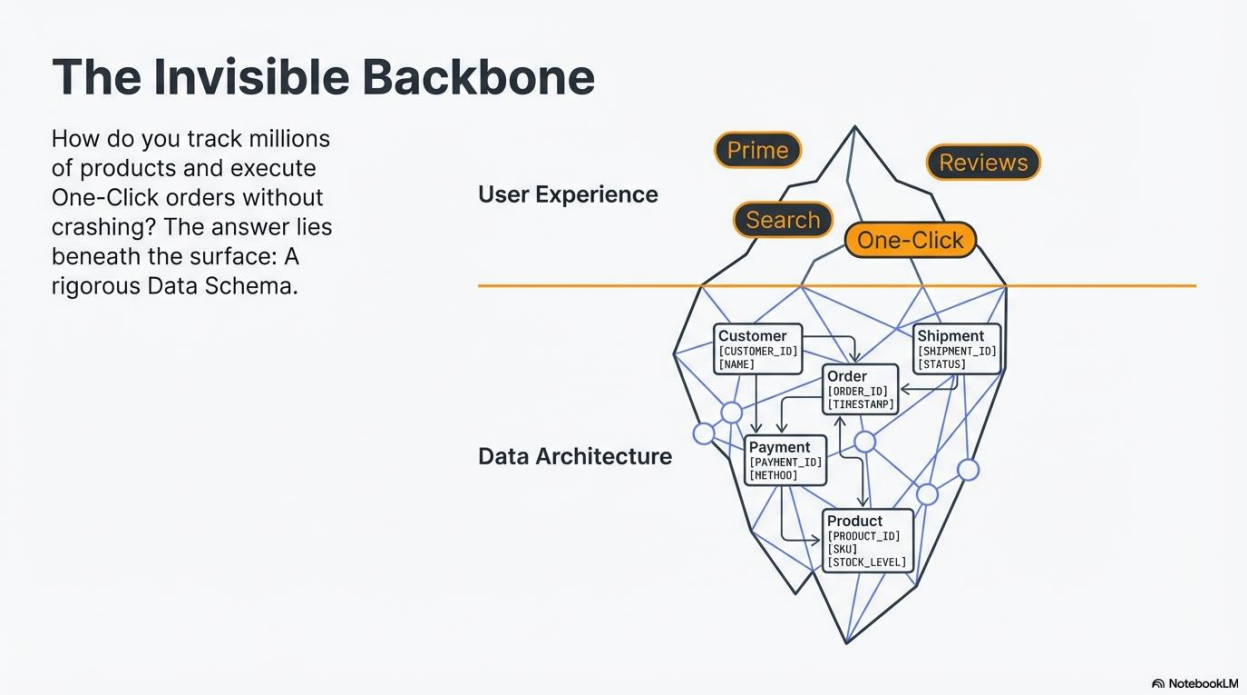
Real-World Challenge: Traditional shopping requires people to spend time traveling to stores, searching for products, and waiting in checkout lines. For busy professionals, elderly people, or those without easy transportation, shopping becomes difficult and time-consuming.

Amazon's Solution:

Amazon provides fast delivery through its network of fulfillment centers located across the country. Amazon Prime members get free two-day delivery on millions of items, and some areas even offer same-day or next-day delivery. Users can shop from home anytime, track their packages in real-time, and have items delivered directly to their doorstep. This saves time and makes shopping much more convenient.

Top Features of Amazon:

- **Vast Product Selection:** Amazon offers millions of products across all categories - electronics, books, clothing, groceries, home goods, and more. Users can find almost anything they need in one place.
- **Customer Reviews and Ratings:** Users can read honest reviews and ratings from other customers who have purchased the product. This helps buyers make informed decisions based on real experiences.
- **Personalized Recommendations:** Amazon suggests products based on browsing history and past purchases, helping users discover items they might be interested in.
- **Amazon Prime:** Prime membership offers free two-day shipping, access to Prime Video streaming, exclusive deals, and other benefits for a yearly subscription fee.
- **One-Click Ordering:** Registered users can complete purchases with a single click, making checkout quick and easy without entering payment details repeatedly.
- **Real-Time Order Tracking:** Users receive updates about their order status and can track packages from the warehouse to delivery at their doorstep.
- **Easy Returns:** Amazon provides simple return processes with prepaid shipping labels, making it risk-free to try new products.
- **Wishlist and Cart:** Users can save items to their wishlist for later consideration or add multiple products to their cart before checking out all at once.
- **Amazon Marketplace:** Third-party sellers can list products on Amazon, giving customers access to even more variety and helping small businesses reach customers.



Schema Rationale

The schema design for Amazon's e-commerce platform is driven by the need to support high-volume transactions, complex user interactions, and scalable operations while maintaining data integrity and clarity. Each entity and relationship is intentionally chosen to mirror real-world e-commerce workflows and business logic.

1. **Customer**

The Customer entity represents registered users on the platform. It is designed as a core entity because almost every business operation—browsing, ordering, payment, and delivery—originates from a customer. Storing customer details separately avoids redundancy and enables one-to-many relationships with orders, carts, and wishlists, reflecting repeat purchase behavior in real-world scenarios.

2. **Product**

The Product entity is central to the platform, as Amazon's primary function is product discovery and sales. Each product has attributes such as price, stock, and description to support inventory visibility and informed purchase decisions. Products are kept independent of orders to allow the same product to be purchased multiple times by different customers.

3. **Category**

The Category entity exists to organize products logically and enhance search and navigation. A one-to-many relationship between Category and Product reflects real-world classification, where each product belongs to a single primary category, while a category

can contain many products. This structure improves scalability and simplifies filtering and analytics.

4. **Cart**

The Cart entity models a temporary, pre-purchase state where customers collect items before checkout. It captures customer intent without committing to a transaction. The relationship between Customer and Cart is one-to-many, allowing customers to add multiple products with varying quantities. Separating Cart from Orders ensures clean handling of abandoned carts and incomplete purchases.

5. **Wishlist**

The Wishlist entity supports long-term user intent and product bookmarking. It is separated from the Cart to distinguish between immediate purchase intent and future consideration. This design enables Amazon to analyse user preferences and drive personalized recommendations without affecting transactional data.

6. **Order**

The Order entity represents a confirmed purchase transaction. It is separated from the Cart to clearly distinguish between intent and commitment. A customer can place multiple orders over time, resulting in a one-to-many relationship between Customer and Order. Storing order-level attributes like order date and status enables tracking of the order lifecycle.

7. **Order_Item**

The Order_Item entity resolves the many-to-many relationship between Orders and Products. An order can contain multiple products, and a product can appear in multiple orders. This entity also stores price_at_purchase, preserving historical pricing accuracy even if product prices change later. This design is critical for financial consistency and auditing.

8. **Payment**

The Payment entity captures transaction-specific details such as payment method, amount, and status. Separating Payment from Order allows flexibility in handling different payment workflows and future extensions such as refunds or retries. The one-to-one relationship reflects the assumption of a single primary payment per order for simplicity.

9. **Shipment**

The Shipment entity manages logistics and delivery tracking. It is separated from Orders to allow independent tracking of dispatch and delivery status. This design supports real-world scenarios where an order may involve complex shipping processes while keeping the order data clean and focused on commerce.

10. **Design Assumptions**

- Each order is associated with a single primary payment.
- Inventory is abstracted as a stock attribute at the product level.
- Address details are simplified for clarity and scope control

---

Schema Description:

The schema for Amazon involves multiple entities that represent different parts of the platform. These entities include Customer, Product, Category, Cart, Wishlist, Order, Order_Item, Payment, and Shipment. Each entity has specific attributes and relationships with other entities.

Customer Entity:

The Customer entity contains information about each user:

- **customer_id (Primary Key):** A unique identifier for each customer.
- **first_name:** The customer's first name.
- **last_name:** The customer's last name.
- **email:** The customer's email address for account communication.
- **Password_hash:** The password for the customer's account.
- **address:** The complete delivery address of the customer.
- **phone_number:** The phone number of the customer.

Product Entity:

Products are the core items available for purchase on Amazon:

- **product_id (Primary Key):** A unique identifier for each product.
- **product_name:** The name of the product.
- **description:** Detailed description of the product.
- **price:** The current price of the product.
- **stock:** The quantity available for purchase.
- **category_id (Foreign Key referencing Category Entity):** The category to which the product belongs.

Category Entity:

Categories help organize products for easy browsing:

- **category_id (Primary Key):** A unique identifier for each category.
- **category_name:** The name of the category (e.g., Electronics, Books, Clothing).

Cart Entity:

Cart stores items that customers plan to purchase:

- **cart_id (Primary Key):** A unique identifier for each cart item.
- **customer_id (Foreign Key referencing Customer Entity):** The customer who added the item to cart.

- **product_id (Foreign Key referencing Product Entity):** The product added to the cart.
- **quantity:** The quantity of the product in the cart.

Wishlist Entity:

Wishlist allows customers to save products for future consideration:

- **wishlist_id (Primary Key):** A unique identifier for each wishlist item.
- **customer_id (Foreign Key referencing Customer Entity):** The customer who saved the item.
- **product_id (Foreign Key referencing Product Entity):** The product saved to the wishlist.

Order Entity:

Order represents purchases made by customers:

- **order_id (Primary Key):** A unique identifier for each order.
- **customer_id (Foreign Key referencing Customer Entity):** The customer who placed the order.
- **order_date:** The date when the order was placed.
- **total_price:** The total amount for the order.
- **order_status:** The current status of the order (Processing, Shipped, Delivered, Cancelled).

Order_Item Entity:

Order_Item represents individual products within an order:

- **order_item_id (Primary Key):** A unique identifier for each order item.
- **order_id (Foreign Key referencing Order Entity):** The order this item belongs to.
- **product_id (Foreign Key referencing Product Entity):** The product being ordered.
- **quantity:** The quantity of the product ordered.
- **price_at_purchase:** The price of the product at the time of purchase.

Payment Entity:

Payment records transaction details for orders:

- **payment_id (Primary Key):** A unique identifier for each payment.
- **order_id (Foreign Key referencing Order Entity):** The order this payment is for.
- **payment_date:** The date when payment was made.
- **payment_method:** The method used for payment (Credit Card, Debit Card, UPI, Cash on Delivery).
- **amount:** The total amount paid.
- **payment_status:** The status of payment (Completed, Pending, Failed).

Shipment Entity:

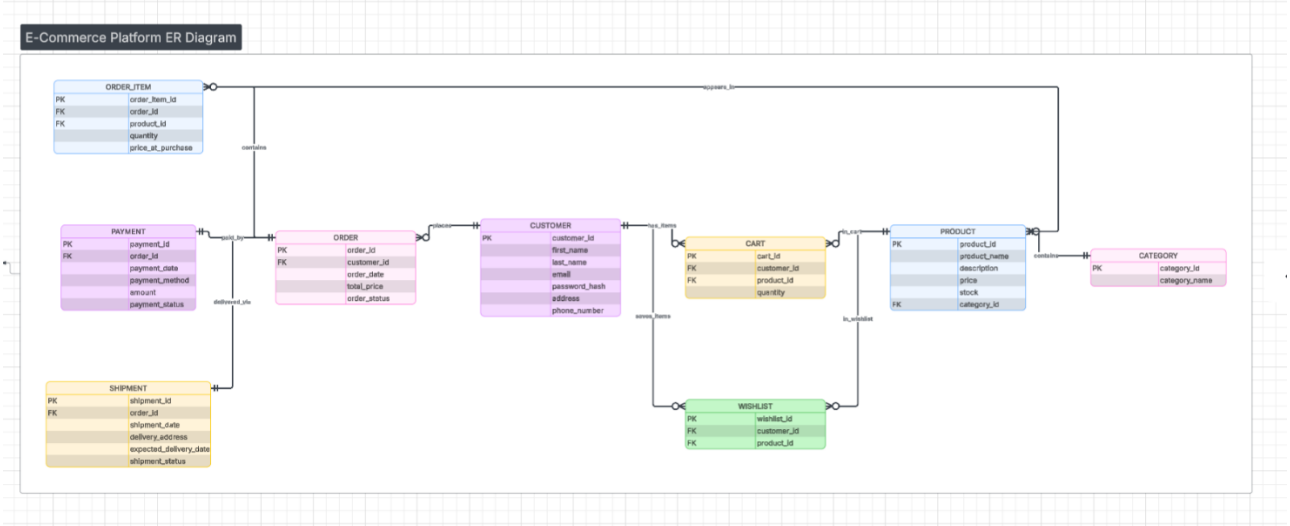
Shipment tracks the delivery of orders:

- **shipment_id (Primary Key):** A unique identifier for each shipment.
- **order_id (Foreign Key referencing Order Entity):** The order being shipped.
- **shipment_date:** The date when the shipment was dispatched.
- **delivery_address:** The address where the order will be delivered.
- **expected_delivery_date:** The estimated delivery date.
- **shipment_status:** The current status of shipment (In Transit, Out for Delivery, Delivered).

Relationships:

- **Customers place Orders:** Each customer can place multiple orders, and each order is placed by a single customer.
- **Orders contain Order_Items:** Each order can have multiple items, and each item belongs to one order.
- **Products belong to Categories:** Each product belongs to a single category, and a category can have multiple products.
- **Payment for Order:** Each order has one payment, and each payment is linked to one order.
- **Shipment for Order:** Each order is linked to one or more shipments, and each shipment belongs to one order.
- **Customers have Cart:** Each customer can add multiple items to their cart, and each cart item belongs to one customer.
- **Customers have Wishlist:** Each customer can save multiple products to their wishlist, and each wishlist item belongs to one customer.
- **Products appear in Orders:** A product can be ordered multiple times by different customers, and each order item references one product.

ER Diagram:



The ER diagram illustrates the core entities involved in Amazon’s e-commerce workflow and their relationships. It shows how customers interact with products through carts and wishlists, place orders, make payments, and receive shipments. The diagram helps visualize the logical structure supporting Amazon’s end-to-end shopping experience.

Conclusion:

In this case study, we examined Amazon's database structure and how different entities work together. Amazon is a comprehensive e-commerce platform that connects customers with millions of products. The system manages users, products, categories, shopping carts, wish lists, orders, payments, and shipments efficiently.

Amazon's well-designed schema enables it to handle millions of transactions daily while providing customers with a smooth shopping experience. The relationships between entities ensure that every step - from product discovery to delivery - works seamlessly. This robust database structure has made Amazon the world's leading e-commerce platform, trusted by customers globally for convenient and reliable online shopping.

This product dissection highlights how a well-structured data model enables Amazon to manage large-scale e-commerce operations efficiently while maintaining a smooth user experience.