What Data Does an E-commerce System Like Amazon Store?

In a comprehensive e-commerce platform like Amazon, various types of data are necessary for its functioning and to provide users with an optimal shopping experience:

1. User/Customer Data: This includes information like names, addresses, and payment details. It forms the core of personalizing the user experience.
2. Product Data: Details like product titles, descriptions, images, and prices form a critical part of the shopping journey.
3. Order Data: Data on order IDs, product IDs, invoice details, item quantities, and delivery/shipping details help in tracking and managing purchases.
4. Inventory Data: Information on stock levels, prices, quantities, and availability ensure accurate product listing and facilitate effective inventory management.
5. User Interaction Data: Capturing user search queries, product behavior, and reviews help in improving user experience and making informed business decisions.

How Do We Choose the Data Store?

Choosing the right data store is a critical aspect of designing an effective e-commerce system. Here are the key factors to consider:

- Data Model: Consider the structure and complexity of your data. A relational database may be a good fit for structured data, whereas NoSQL databases can handle more complex, unstructured data better.
- Scalability: Look at the system's throughput, latency, and volume requirements. Will your data store be able to scale as these grow?
- Consistency: Consistency is crucial for maintaining accurate and current data. Choose a data store that provides the right level of consistency for your needs.
- Availability: Your data store should be highly available to prevent any downtime that could impact the user experience.
- Cost: Keep in mind the costs associated with setting up and maintaining the data store.
- Performance Requirements: Identify whether your system is read-heavy or write-heavy, and select a data store that performs best under those conditions.
- Data Security & Compliance: Your data store must meet all necessary security standards and compliance requirements.
- Community and Support: The availability of robust community support can be invaluable when troubleshooting issues.
- Future Requirements: Always keep an eye on the future. As your system evolves, so will your data storage needs.

Different Types of Data Stores

Type	Subtype	Description	Examples
Database	Relational	Organizes data into tables. Good for structured data and complex queries.	MySQL, PostgreSQL, OracleDB, SQLite, MariaDB
	Non-Relational - Document Stores	Stores data as documents. Ideal for storing semi-structured data.	MongoDB, CouchDB, RavenDB
	Non-Relational - Key-Value Stores	Stores data as a collection of key-value pairs. Very fast and simple.	Redis, DynamoDB, Riak
	Non-Relational - Columnar Databases	Stores data by columns instead of rows. Ideal for analytics and big data.	Apache Cassandra, HBase
	Non-Relational - Graph Databases	Stores data as nodes and edges. Suitable for interconnected data.	Neo4j, Amazon Neptune
	Non-Relational - Time Series Databases	Optimized for time-stamped data. Suitable for IoT, telemetry, etc.	InfluxDB, OpenTSDB
Data Store	In-Memory Databases	These databases store all their data in the main memory (RAM) of the server, offering very high speed and low latency, ideal for caching and real-time applications.	Redis, Memcached
	Distributed File Systems	These are file systems that allow data to be stored across multiple machines but accessed and manipulated like it's on one. They are great for big data applications.	Hadoop Distributed File System (HDFS), Google File System (GFS)

	Content Delivery Network (CDN)	CDNs are a globally distributed network of servers that provide fast delivery of internet content. They cache static resources closer to users for improved performance.	Cloudflare, Akamai, Amazon CloudFront
	Data Warehouses	These are large repositories of data collected from different sources, designed to support business intelligence activities, particularly analytics. Data is consolidated, transformed and stored at a granular level.	Google BigQuery, Amazon Redshift, Snowflake
	Message Brokers	These are tools that receive incoming data from applications and send it to different applications for processing. They act as a buffer for incoming data and can help manage and process large amounts of data where delivery time is not a concern.	Apache Kafka, RabbitMQ, Google Pub/Sub

SQL vs NoSQL

	SQL Databases	**NoSQL Databases**
Data Model	Relational, data is structured and organized into tables.	Non-relational, can handle structured, semi-structured, and unstructured data.
Consistency	ACID Compliance Strong Consistency	NoSQL databases may offer strong consistency or eventual consistency. Async
Schema	Fixed schema. Data must adhere to defined structure.	Schema-less. Provides flexibility to store diverse data models (key-value, document, column, graph).
Scalability	Scale vertically by increasing the horsepower (CPU, RAM, SSD) of an existing server.	Scale horizontally by adding more servers to handle more traffic.
Transactions	Supports ACID (Atomicity, Consistency, Isolation, Durability) transactions.	Many do not support ACID transactions, though there are exceptions like MongoDB. Typically, they provide eventual consistency.
Query Language	Use Structured Query Language (SQL) which is powerful for complex queries.	No standard query language. Methods for data manipulation vary.
Examples	MySQL, PostgreSQL, Oracle Database.	MongoDB, Apache Cassandra, Redis, Amazon DynamoDB.