Section A: Limitations of RDBMS

Relational Database Management Systems (RDBMS) face several challenges when handling modern, dynamic data requirements:

- Different Product Attributes: RDBMS relies on fixed tables with predefined columns. If products vary widely (e.g., laptops with RAM/processor vs. shoes with size/color), designing a single schema becomes inefficient. Many columns remain unused, leading to sparse data and wasted storage.

- Frequent Schema Changes: Adding new product types often requires altering the schema (e.g., adding new columns). Schema modifications are costly, time-consuming, and can disrupt existing applications. This rigid structure makes RDBMS unsuitable for rapidly evolving product catalogs.

- Nested/Hierarchical Data: Customer reviews often include nested elements like ratings, comments, replies, or metadata. RDBMS stores data in flat, tabular form, making it difficult to represent hierarchical or JSON-like structures without complex joins or separate tables. This reduces query efficiency and complicates application logic.

In short, RDBMS struggles with flexibility, scalability, and handling semi-structured data.

Section B: NoSQL Benefits

MongoDB, a leading NoSQL database, addresses the limitations of traditional RDBMS through its flexible design:

- **Flexible Schema**: Unlike rigid relational tables, MongoDB uses a document-oriented model (JSON/BSON). Each product can have its own attributes without forcing unused columns. For example, laptops can store RAM and processor fields, while shoes can store size and color, all within the same collection.
- **Embedded Documents**: MongoDB allows nesting documents inside others. Customer reviews can be stored directly within the product document, including ratings, comments, and replies. This eliminates complex joins, making queries faster and data more naturally aligned with real-world structures.
- **Horizontal Scalability**: MongoDB supports sharding, distributing data across multiple servers. As product catalogs and customer reviews grow, the database scales horizontally, ensuring high availability and performance without the bottlenecks of vertical scaling in RDBMS.

In short, MongoDB provides flexibility, efficiency, and scalability, making it ideal for dynamic product catalogs and user-generated content.

Section C: Trade-offs

Two disadvantages of using MongoDB instead of MySQL for this product catalog

MongoDB supports transactions but they are more complex and less mature. For an e-commerce catalog where consistency across multiple tables (e.g., stock, orders, payments) is critical, MongoDB may introduce risks of partial updates or data inconsistency.

□ MongoDB lacks native joins (though it has $lookup), making cross-product queries or aggregations harder to design and potentially slower. For example, generating sales reports across diverse product categories may require complex aggregation pipelines instead of straightforward SQL queries.