DBD381 Project

Belgium campus itversity

Research section

Gabriella Petersen

TRENT EVANS   
LUQMAAN SLARMIE

Xavier Barnard

2025

Contents

[Overview of NoSQL Technologies: 2](#_Toc198818440)

[1. MongoDB 2](#_Toc198818441)

[2. Cassandra 3](#_Toc198818442)

[3. Redis 3](#_Toc198818443)

[E-Commerce Suitability Breakdown 4](#_Toc198818444)

[Real-World E-Commerce Examples 4](#_Toc198818445)

[Selected NoSQL Database: MongoDB 5](#_Toc198818446)

[Justification for Selection 5](#_Toc198818447)

[Polyglot Persistence: The Best of Both Worlds 5](#_Toc198818448)

[Future Research and Applications 6](#_Toc198818449)

[References (temporary): 6](#_Toc198818450)

# Overview of NoSQL Technologies:

NoSQL (Not Only SQL) databases have gained significant traction in modern application development due to their ability to offer flexibility, scalability, and high-speed data processing. This research focuses on three prominent NoSQL databases — MongoDB, Cassandra, and Redis — evaluating their architectural models, strengths, limitations, and their suitability for e-commerce applications.

## 1. MongoDB

* Type: Document-Oriented (JSON-like documents)
* Strengths:
  + Highly flexible schema design, ideal for evolving data structures such as dynamic product catalogues
  + Native support for horizontal scalability through sharding
  + Advanced capabilities in indexing, complex querying, and data aggregation
  + Backed by a robust, active community and a rich ecosystem of tools
* Weaknesses:
  + May experience performance bottlenecks under extremely write-intensive workloads
  + Full ACID-compliant transactions were historically limited (though recent versions have addressed this)

## 2. Cassandra

* Type: Wide-Column Store
* Strengths:
  + Exceptional scalability and fault tolerance with no single point of failure
  + Optimized for high-throughput, write-heavy applications
  + Decentralized architecture ensures equal responsibility among all nodes
* Weaknesses:
  + Less flexible querying capabilities; data modelling must align closely with expected query patterns
  + Configuration and performance tuning can be complex

## 3. Redis

* Type: In-Memory Key-Value Store
* Strengths:
  + Extremely low-latency performance due to in-memory architecture
  + Ideal for caching, real-time analytics, leaderboard systems, and session storage
  + Simple data structures and straightforward integration
* Weaknesses:
  + Limited suitability as a primary data store for large-scale datasets
  + Data persistence is optional, posing potential risks if not configured properly

# E-Commerce Suitability Breakdown

| **Feature** | **MongoDB** | **Cassandra** | **Redis** |
| --- | --- | --- | --- |
| **Type** | Document-oriented | Wide-column store | Key-value (in-memory) |
| **Scalability** | High (via sharding) | Extreme (linear, no single point of failure) | Moderate (limited by memory) |
| **Performance** | Balanced (read/write friendly) | Optimized for write-heavy workloads | Extremely fast (in-memory speed) |
| **Data Model** | Flexible JSON-like documents | Denormalized wide-column tables | Simple key-value pairs |
| **Querying Flexibility** | High | Limited | Very limited |
| **Best Use Case** | General-purpose e-commerce | Order/log-heavy systems | Real-time features (cache, session, leaderboard) |

# Real-World E-Commerce Examples

* Amazon utilizes multiple types of databases for different services. While their product catalogue might live in a flexible document-based or wide-column store, their recommendation engine may use an in-memory solution like Redis.
* eBay uses MongoDB for handling their metadata storage because of its ability to evolve schemas rapidly and manage massive scale.
* Alibaba combines distributed systems with NoSQL databases like Cassandra to handle real-time messaging and transactions at scale.

These platforms exemplify the core idea of polyglot persistence, where multiple databases are used together to handle different workloads more efficiently.

# Selected NoSQL Database: MongoDB

## Justification for Selection

MongoDB has been identified as the most suitable NoSQL solution for the intended e-commerce application, offering the best balance of flexibility, scalability, and robust query support.

* Document-Oriented Data Model: MongoDB’s schema-less document structure is naturally aligned with the needs of an e-commerce platform. Entities such as users, products, orders, and reviews can be modelled as standalone, richly structured JSON documents.
* Schema Flexibility: New data elements — for example, "flash sale tags" or "customer wish lists" — can be added without requiring a schema redesign, thereby supporting rapid development and feature deployment.
* Scalable Architecture: With native support for sharding and replication, MongoDB can seamlessly handle high-traffic events such as seasonal sales or promotional campaigns.
* Rich Query Capabilities: MongoDB supports advanced filtering and aggregation, allowing for complex queries such as retrieving highly rated products purchased within a specific timeframe.

While Cassandra and Redis offer exceptional strengths in specific areas — high-volume writes and real-time speed respectively — MongoDB is better suited for handling diverse and evolving requirements typical of a modern e-commerce ecosystem.

# Polyglot Persistence: The Best of Both Worlds

While MongoDB will serve as the primary database, integrating Redis as a secondary, in-memory layer can enhance real-time functionality. Example implementations:

* Cache high-demand product pages in Redis for lightning-fast access.
* Store user sessions and shopping carts in Redis to reduce latency.
* Run real-time analytics like active user counters or trending product lists.

This hybrid setup improves performance without compromising long-term data durability.

# Future Research and Applications

* Multi-Cloud Deployment: Exploring MongoDB deployments across multiple cloud regions to enhance availability, compliance, and disaster recovery.
* AI/ML Integration: Leveraging MongoDB as a data foundation for machine learning models that provide personalized product recommendations based on user behaviour.
* MongoDB Atlas Adoption: Utilizing MongoDB’s fully managed cloud platform for streamlined database operations, including automated scaling, performance monitoring, and backups.
* Hybrid Architecture with Redis: Implementing Redis alongside MongoDB to cache high-demand product pages and sessions, combining persistent data storage with ultra-fast access.

# References (temporary):

* Lakshman, A., & Malik, P. (2010). Cassandra: A decentralized structured storage system. \*ACM SIGOPS Operating Systems Review, 44\*(2), 35–40. https://doi.org/10.1145/1773912.1773922
* MongoDB, Inc. (2023). \*What is MongoDB?\*. https://www.mongodb.com/what-is-mongodb
* Redis Ltd. (2023). \*What is Redis?\*. https://redis.io/docs/about/
* Stonebraker, M. (2018). \*SQL vs. NoSQL: What’s the difference?\* ACM Queue, 16(3), 1-10. https://doi.org/10.1145/3201287.3201291
* Sadalage, P. J., & Fowler, M. (2012). \*NoSQL distilled: A brief guide to the emerging world of polyglot persistence\*. Addison-Wesley.