Universidad Distrital
Francisco Jose de Caldas

Database Management System Report for E-Commerce.

Luis Sebastian Martinez Guerrero

Leidy Marcela Morales Segura

Supervisor: Carlos Andres Sierra

May 15, 2025

Abstract

This report presents a comprehensive analysis and design of a database management system tailored to the needs of an e-commerce platform similar to Mercado Libre. The goal is to achieve high availability, fast performance, secure transactions, and business intelligence integration for concurrent users. Using an architecture based on read/write separation, caching, and replication, the project addresses the critical challenges of scalability, fault tolerance, and real-time analytics in a big data environment.

Keywords: E-Commerce, Data Base Design, Distributed Architecture, ER Model

Report's total word count: 3131

Contents

List of Figures

List of Tables

List of abbreviations

e-Commerce	electronic commerce
BI	business intelligence
SQL	Structured Query Language
DBMS	Database Management System
API	Application Programming Interface
ER	Entity-Relationship
NoSQL	Not Only SQL

Chapter 1

Introduction

In today's dynamic digital economy, the demand for robust, scalable, and intelligent e-commerce platforms has become increasingly essential. E-Commerce has evolved from simple online stores to complex ecosystems capable of managing millions of simultaneous transactions, personalized recommendations, and real-time business analytics. The foundation of these platforms lies in the need for a high-performance database management system capable of managing diverse types of data (structured, semi-structured, and unstructured) across multiple regions with minimal latency. This project focuses on the design of a database management system for an e-commerce platform similar to Mercado Libre, one of the leading e-commerce platforms in Latin America.

Mercado Libre's infrastructure must support a wide range of services: user authentication, product listings, transaction processing, inventory updates, customer reviews, and promotions. All of these functionalities require a well-integrated backend, supported by data systems that guarantee availability, consistency, partition tolerance, and rapid response. Traditional monolithic database solutions are no longer sufficient for such complex and demanding scenarios. The transition to distributed architectures where data is partitioned, replicated, and processed across geographically dispersed nodes offers significant advantages in terms of reliability, performance, and scalability.

This paper presents a database design and management proposal that reflects the real needs of an e-commerce system. It includes a detailed set of functional and non-functional requirements derived from user stories and system expectations. Among the main challenges addressed are the need for high availability, rapid query responses, seamless data ingestion, real-time analytics, and secure data management. These aspects are critical to providing an optimal user experience and providing administrators, business analysts, and decision-makers with accurate and timely information.

To achieve these goals, the proposed system employs a database architecture that separates reads (queries) from writes (transactions), uses caching, and utilizes asynchronous replication. Business intelligence is integrated through data warehousing and visualization tools, enabling strategic insights from transactional and behavioral data.

This introduction lays the groundwork for understanding the broader scope of the high-level database project. It provides the context, motivation, and technological justification for each decision and defines the roadmap for a system designed not only to meet current business demands but also to adapt to future growth and innovation.

1.1 General Objective

To design and document a distributed database system for an e-commerce platform that ensures performance, security, and business insight.

1.2 Specific Objective

- Define functional and non-functional requirements.
- Establish a scalable and fault-tolerant architecture.
- Design an entity-relationship model and normalize the data.
- Implement support for role-based access and security.
- Enable integration with BI tools for strategic decision-making.

1.3 Background

The evolution of database technologies has reflected the exponential growth and increasing complexity of modern e-commerce platforms. Initially, most systems operated with monolithic databases, centralized, and vertically scalable engines designed for smaller, isolated applications. These systems were sufficient during the early stages of digital commerce, offering simple configurations and easy control. However, as user bases expanded and transaction volumes skyrocketed, the limitations of monolithic architectures became increasingly apparent. These included data processing bottlenecks, the inability to scale horizontally, and difficulties supporting multi-regional operations and real time analytics.

In contrast, modern e-commerce ecosystems demand distributed, resilient, and intelligent database infrastructures. These systems must handle high-throughput workloads, enable rapid query performance, and support concurrent access from millions of users globally. The shift from monolithic databases to distributed systems addresses these challenges by introducing capabilities such as horizontal scaling, data sharding, in-memory caching, and asynchronous replication.

Mercado Libre, a leading e-commerce platform in Latin America, exemplifies this technological evolution. Initially built on traditional relational databases, the company has progressively adopted advanced data infrastructure strategies. Today, its architecture leverages a combination of microservices, distributed NoSQL and SQL databases, real-time data streaming (e.g., Apache Kafka), and cloud-native platforms to manage and scale its operations. These systems support vital platform features such as fraud detection, product recommendations, logistics optimization, and business intelligence.

A relevant comparison between traditional and current database models can be made in terms of scalability, fault tolerance, and analytics capability. While monolithic databases rely on vertical scaling and are prone to single points of failure, Mercado Libre's distributed system uses multi-region replication and elastic scalability, thus improving availability and performance.

Recent advancements in unified architectures like the Lakehouse model Zaharia et al. (2021) further support the integration of operational and analytical workloads. This model overcomes the limitations of siloed data systems by combining the reliability of data warehouses with the scalability of data lakes. Such solutions are increasingly adopted in large-scale e-commerce environments to enable faster, more intelligent decision-making.

In conclusion, the transition from monolithic databases to distributed, high-performance platforms has become essential for e-commerce scalability and competitiveness. The case of Mercado Libre demonstrates how the adoption of modern data architectures supports innovation, enhances user experience, and enables continuous operational efficiency in a data-intensive environment.

1.4 Scope

The scope of this project encompasses the design and documentation of a distributed database management system tailored for a large-scale e-commerce platform, with Mercado Libre. The system includes key functionalities such as user registration and secure authentication, product and inventory management, transaction and order tracking, and personalized recommendation engines. Additionally, it supports promotional campaign configuration, real-time data replication across multiple geographic regions, and caching strategies to reduce latency. Emphasis is placed on the integration of business intelligence tools that allow for dynamic reporting and data visualization to support managerial decision-making. The system also incorporates security features including role-based access control and encryption, ensuring compliance with data protection regulations. Designed under the principles of scalability, availability, and performance, the architecture leverages technologies such as PostgreSQL and Redis to meet current and future operational demands. The project remains within the conceptual and design phases, serving as a foundational blueprint for implementing a scalable, resilient, and intelligent database system aligned with the needs of modern e-Commerce.

1.5 Assumptions

During the research and design of the distributed database system for the e-Commerce platform, the following assumptions were made:

- Data Volume and Growth: The platform is assumed to process millions of transactions and user interactions per day, requiring storage and processing capabilities to scale adequately.
- Real-Time Requirements: The system must support real-time recommendations and analytics.
- User Behavior and Access Patterns: User activity is assumed to follow typical e-commerce patterns and that a global user base accesses the platform simultaneously.
- Tools: The system assumes the use of widely adopted technologies such as PostgreSQL and Redis. Considering the project's evolution to other phases, the technologies necessary to meet the objectives will be evaluated.
- Security Context: Role-based access control is assumed.

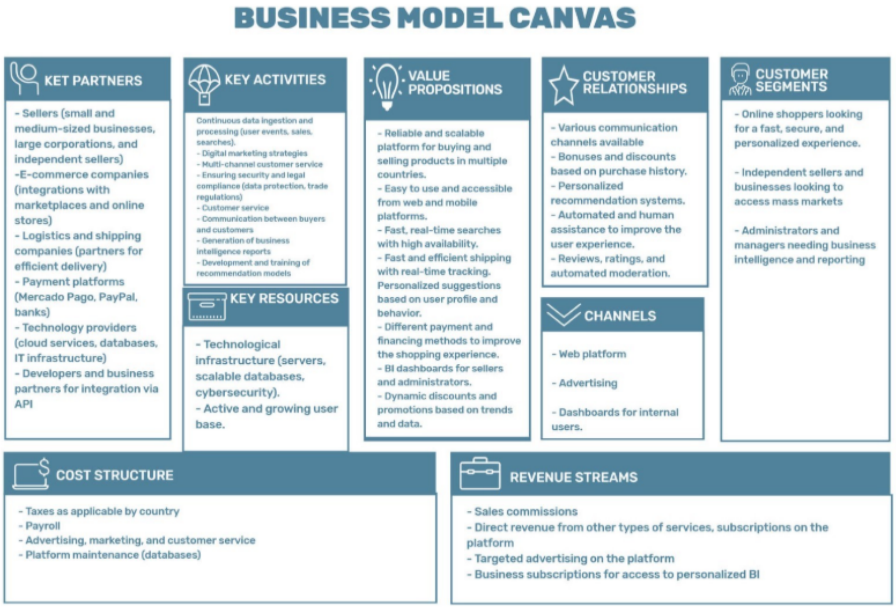
- Data quality and integrity: Input data (e.g., product listings, user reviews) is assumed to be mostly valid, and data validation mechanisms will handle exceptions.

- Business scope: Mercado Libre's business model and operations serve as a reference, but integration with external APIs, payment processors, and logistics partners is excluded.

1.6 Business Model Canvas

To better understand the strategic vision and key components of the e-commerce platform, a Business Model Canvas was developed in figure 1.1. This model provides a comprehensive view of how the system creates, delivers, and captures value across its ecosystem of users, partners, and internal stakeholders.

The inclusion of this model facilitates alignment between the system's technical design and broader business objectives, helping to ensure that the proposed architecture addresses not only performance and scalability needs but also the economic, operational, and customer-facing aspects of the platform. It highlights essential elements such as key activities, resources, partners, customer segments, value propositions, and revenue models.



BUSINESS MODEL CANVAS

KET PARTNERS
- Sellers (small and medium-sized businesses, large corporations, and independent sellers)
-E-commerce companies (integrations with marketplaces and online stores)
- Logistics and shipping companies (partners for efficient delivery)
- Payment platforms (Mercado Pago, PayPal, banks)
- Technology providers (cloud services, databases, IT infrastructure)
- Developers and business partners for integration via API

KEY ACTIVITIES
Continuous data ingestion and processing (user events, sales, searches).
- Digital marketing strategies
- Multi-channel customer service
- Ensuring security and legal compliance (data protection, trade regulations)
- Customer service
- Communication between buyers and customers
- Generation of business intelligence reports
- Development and training of recommendation models

KEY RESOURCES
- Technological infrastructure (servers, scalable databases, cybersecurity).
- Active and growing user base.

VALUE PROPOSITIONS
- Reliable and scalable platform for buying and selling products in multiple countries.
- Easy to use and accessible from web and mobile platforms.
- Fast, real-time searches with high availability.
- Fast and efficient shipping with real-time tracking.
Personalized suggestions based on user profile and behavior.
- Different payment and financing methods to improve the shopping experience.
- BI dashboards for sellers and administrators.
- Dynamic discounts and promotions based on trends and data.

CUSTOMER RELATIONSHIPS
- Various communication channels available
- Bonuses and discounts based on purchase history.
- Personalized recommendation systems.
- Automated and human assistance to improve the user experience.
- Reviews, ratings, and automated moderation.

CHANNELS
- Web platform
- Advertising
- Dashboards for internal users.

CUSTOMER SEGMENTS
- Online shoppers looking for a fast, secure, and personalized experience.
- Independent sellers and businesses looking to access mass markets
- Administrators and managers needing business intelligence and reporting

COST STRUCTURE
- Taxes as applicable by country
- Payroll
- Advertising, marketing, and customer service
- Platform maintenance (databases)

REVENUE STREAMS
- Sales commissions
- Direct revenue from other types of services, subscriptions on the platform
- Targeted advertising on the platform
- Business subscriptions for access to personalized BI

Figure 1.1: Business Model Canvas

Chapter 2

Literature review

Today, e-commerce platforms like Mercado Libre manage large volumes of data from user interactions, transactions, and market trends. Efficient management of this data is critical to improving the user experience, optimizing logistics, and increasing business profitability. Academic and technical literature supports the integration of distributed systems, real-time analytics, and big data technologies as enablers of these capabilities.

The article Yaqoob et al. (2016) provides a comprehensive overview of the evolution of big data and the technologies that have driven its growth. Since its emergence, big data has transformed the way businesses store, process, and analyze information, enabling real-time data-driven decision-making. In the e-commerce sector, the integration of big data technologies such as batch processing, real-time streaming, data mining, and machine learning has improved product recommendation systems, fraud detection mechanisms, and personalized customer experiences.

Furthermore, the work Alrumiah and Hadwan (2021)explores how the application of big data analytics in e-commerce benefits both suppliers and consumers. According to their findings, big data facilitates more informed business decisions, strengthens customer loyalty, and enables large-scale personalization. However, they also highlight the challenges posed by the rapid increase in data volume and associated processing costs, which must be addressed to achieve optimal impact.

However,Agarwal (2025) emphasize the importance of resilience in database infrastructure. They analyze fault-tolerance strategies such as data replication, consensus protocols, and distributed ledgers, which are essential for maintaining availability and consistency in systems such as those used by Mercado Libre. By applying these principles, e-Commerce platforms can ensure service continuity even in the face of failures.

Furthermore, recent studies underscore the role of robust infrastructure and scalable solutions, particularly cloud computing, in the efficient management of massive data sets. Platforms like Mercado Libre rely on these technologies to manage millions of daily transactions, reduce operational risk, and increase performance on scale.

The growing need for big data solutions in e-Commerce emphasizes the importance of designing database architectures capable of managing information quickly, efficiently, and resiliently. This technological foundation is crucial to maintaining innovation and competitiveness in the digital commerce sector.

Chapter 3

Methodology

This research and design project followed an iterative and structured methodology based on the Agile Scrum framework, which combines software engineering practices and database architecture principles. The objective was to ensure a replicable, transparent, and robust design process for a distributed database system adapted to a modern e-Commerce platform. The adoption of Scrum enabled flexible planning, incremental delivery of results, and continuous adaptation to requirements and feedback obtained during project iterations.

- Requirements Analysis: The study began with the identification of user roles (e.g., buyer, seller, administrator, business intelligence analyst, database administrator, platform manager, marketing specialist) and associated functionalities. Functional and non-functional requirements were extracted from user stories and business needs.

- Requirements Analysis: The study began with the identification of user roles (e.g., buyer, seller, administrator, business intelligence analyst, database administrator, platform manager, marketing specialist) and associated functionalities. Functional and non-functional requirements were extracted from user stories and business needs.

- Data Modeling: A conceptual data model was developed using entity-relationship (E-R) diagrams. Normalization up to third normal form (3NF) was applied to reduce redundancy and ensure referential integrity. This model served as the basis for schema generation and system modularization.

- Architectural Design: A layered architecture was proposed. PostgreSQL was selected for transactional consistency, Redis for caching, and Apache Kafka for data ingestion and streaming. The architecture separates write- and read-intensive workloads and integrates asynchronous replication for fault tolerance and high availability. It also includes a cache for handling temporary promotions.

- Analysis of design alternatives: The advantages and disadvantages between relational and NoSQL systems, vertical vs. horizontal scaling, and batch vs. streaming analytics were evaluated to guide component selection and integration.

- Business intelligence integration: Visualization tools are considered to support decision-making.

- Collaboration and Tools: The project uses collaborative tools such as shared documents in the cloud and version control on GitHub to ensure project progress and collaborative work.

- This methodology provides a solid foundation for replicating or extending the proposed system in real-world development environments.

Chapter 4

Results

- As a result of the iterative research and development process, a database management system for an e-commerce company was designed. The project defined 31 functional and 12 non-functional requirements, covering key features such as product management, user interactions, transactional workflows, promotions, and business intelligence integration. The proposed functional and non-functional requirements are presented below.

Table 4.1: Functional Requirements of the E-commerce Platform

ID	Requirement Name	Description	Required Inputs (Fields)
RF01	User Registration	The system must allow user registration with secure authentication.	Full Name, Email, ID Number, Address, Phone Number, Password
RF02	Product Search	The system must allow product searches using quick filters (category, price, rating).	Keyword, Category, Minimum Price, Maximum Price, Minimum Rating
RF03	Personalized Recommendations	The system must display recommendations based on user behavior.	User ID, Browsing History, Purchase History
RF04	Multiple Payment Methods	The system must allow payments via card, transfer, or cash on delivery.	Order ID, Payment Method, Card/Bank Info, Shipping Address
RF05	Publish Reviews	The system must allow reviews to be published only for purchased products.	User ID, Product ID, Rating, Comment
RF06	Create Listings	The system must allow products to be published with details and images.	Product Name, Description, Price, Stock, Images, Category
RF07	Product Statistics	The system must display statistics on views, sales, reviews, and stock.	Seller ID, Product ID, Date Range
RF08	Low Stock Alerts	The system must send notifications when stock is low.	Product ID, Threshold, Current Stock

(Continued on next page)

ID	Requirement Name	Description	Required Inputs (Fields)
RF09	Promotion Management	The system must allow creation of promotions with rules and duration.	Products, Discount, Start Date, End Date, Conditions
RF10	QA Management	The system must allow management of questions and answers in listings.	Product ID, Question ID, Response Text
RF11	Cluster Monitoring	The system must monitor node status and alerts.	System Logs, Node Metrics, Alert Thresholds
RF12	Multi-Region Replication	The system must allow data replication across regions with low latency.	Regions, Cluster Configuration
RF13	Automatic Backups	The system must schedule and store backups.	Date/Time, Backup Type, Destination
RF14	Query Optimization	The system must optimize response time for frequent queries.	Frequent Queries, Fields for Indexing/Partitioning
RF15	Category Management	The system must allow creation/modification of categories visible in filters.	Name, Parent Category, Description
RF16	Metrics by Region and Category	The system must display segmented key performance indicators.	Date, Region, Category, Metric Type
RF17	Content Moderation	The system must allow moderation of listings and reviews with traceability.	Content ID, Action, Justification, Moderator ID
RF18	System Error Notification	The system must alert the administrator in case of system errors.	Error Type, Affected Service, Date/Time, Severity
RF19	Sales KPIs Dashboard	The system must display key sales indicators in real time.	Date, Region, Category, KPI
RF20	Category/Region Performance	The system must visualize segment comparisons.	Date, Category, Region, Metric Type
RF21	Recommendation Effectiveness Report	The system must measure the effectiveness of personalized recommendations.	User ID, Recommended Products, Interactions, Purchases
RF22	Query Efficiency	The system must evaluate the performance and frequency of system queries.	Query Name, Execution Time, Frequency

End of table

Table 4.2: Non-Functional Requirements of the E-commerce Platform

ID	Requirement Name	Description
NFR01	Big Data	The system must process and store large volumes of structured and unstructured data in both real-time and batch modes using centralized repositories for analysis.
NFR02	Fast Query Response	The system must ensure low latency for read/write operations using optimized transactional databases, in-memory caching, and advanced indexing for complex queries.
NFR03	Data Ingestion	The system must support massive batch ingestion and real-time streaming with fault tolerance and the ability to reprocess data when needed.
NFR04	Business Intelligence	The system must integrate with analytics tools and provide real-time information for dashboards to strategic insights.
NFR05	Multi-Location Access	The system must provide global access through automatic geographic replication, proximity-based load balancing, and efficient delivery of static content to reduce latency.
NFR06	Recommendation System	The system must dynamically generate personalized recommendations, maintain user and product profiles, and update them in real-time based on recent interactions.
NFR07	High Availability	The system must ensure 24/7 operation using automatic data replication, disaster recovery mechanisms (failover), and continuous performance monitoring.
NFR08	Scalability	The system must adapt to variable demand using horizontal scaling, data sharding, and elastic architecture with auto-scaling of resources.
NFR09	Security	The system must provide data encryption at rest and in transit, implement role-based access control (RBAC), and audit critical operations to prevent vulnerabilities.
NFR10	Regulatory Compliance	The system must comply with data privacy regulations and industry standards, including payment security certifications.
NFR11	Maintainability	The system must be modular to allow updates without interruption, include detailed technical documentation, and support automated deployment and monitoring.
NFR12	Cost Optimization	The system must optimize infrastructure usage through auto-scaling, cold storage for historical data, and strategies to avoid overprovisioning.
		End of table

- An entity-relationship (ER) model was created and normalized to third normal form, optimizing the data structure for scalability and consistency as shown in the figure 4.1

- The proposed architecture incorporates technologies such as PostgreSQL for transactional storage, Redis for frequently queried caching, and Apache Kafka for real-time data ingestion. The architecture also supports asynchronous replication, read-write sep-

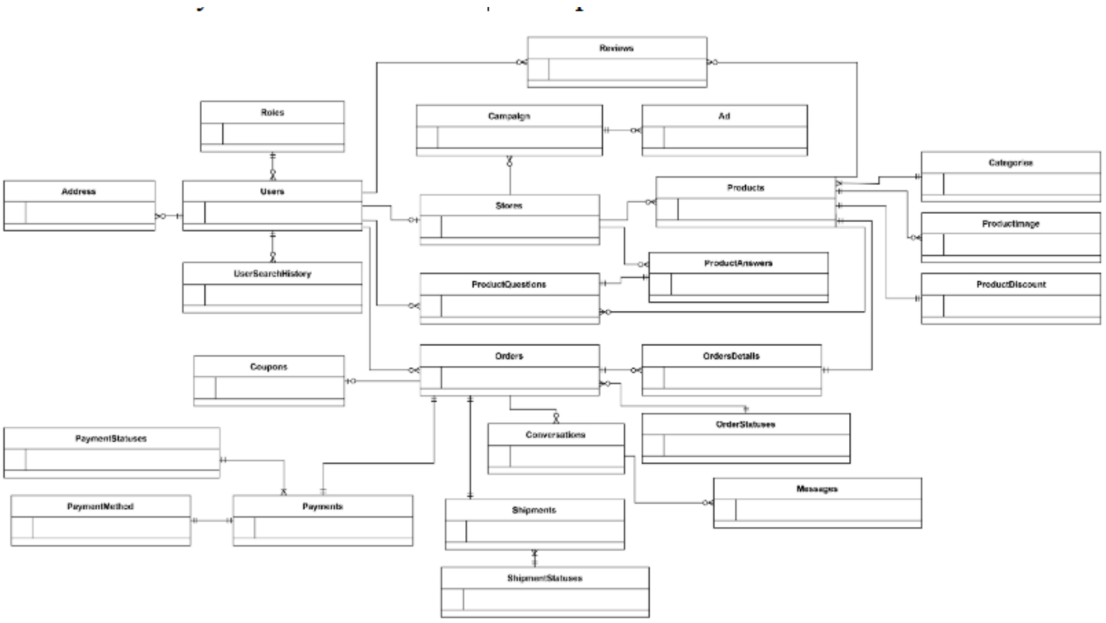


Figure 4.1: Entity-relationship model

aration, and multi-region access, enabling high availability and fault tolerance, as shown in figure 4.2.

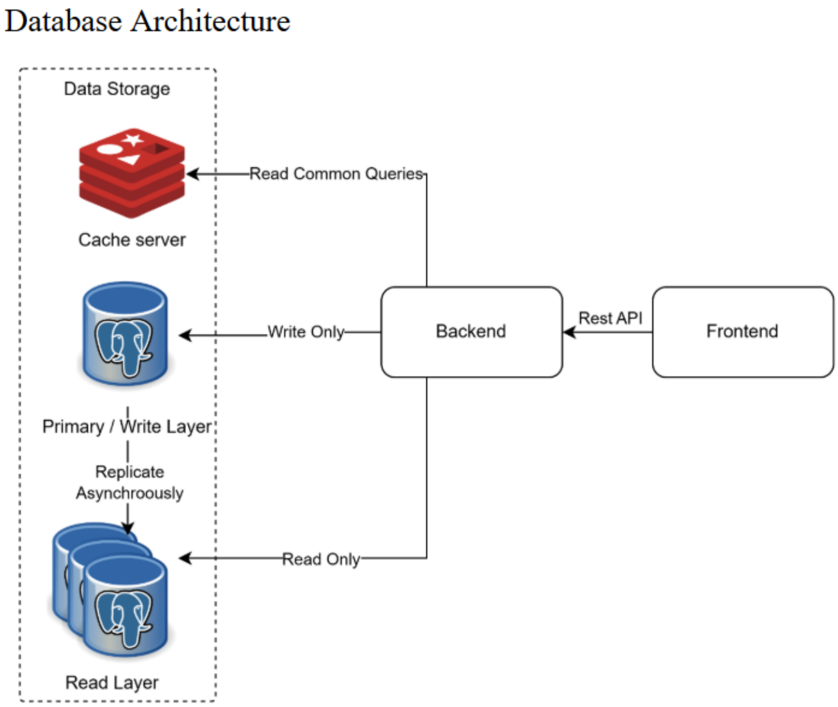


Figure 4.2: Architecture database

- Security measures such as role-based access control, data encryption, and audit logging were incorporated into the system design. Business intelligence functionality is supported by defining ETL pipelines and reporting dashboards tailored to different roles, such as marketing professionals, analysts, and administrators.

- The project also generated user stories for seven key roles proposed in figure 4.3 and mapped them to system requirements, providing traceability and clarity. The implementation and testing phase will be completed in the next phase of the course. Detailed documentation and architectural diagrams provide a solid foundation for future development and implementation.

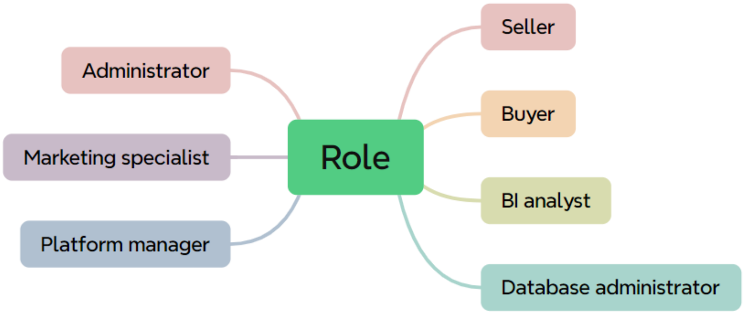


Figure 4.3: User Roles

Chapter 5

Conclusions

This project demonstrates the feasibility and relevance of designing a distributed database management system to meet the demands of a modern e-commerce platform. By addressing both business and technical perspectives, the design ensures support for high-volume transactions, real-time recommendations, and business intelligence with a scalable and secure infrastructure.

The integration of PostgreSQL and Redis in a layered architecture addresses essential issues such as consistency, latency, and fault tolerance. Functional and non-functional requirements were derived from user stories aligned with realistic roles, making the system adaptable and aligned with operational realities.

The proposal's key strengths include multi-region replication, asynchronous communication, secure role-based access, and strategic support for analytics using business intelligence dashboards. Furthermore, alignment with the Business Model Canvas ensures the system contributes to broader business objectives such as user retention, efficient logistics, and monetization strategies.

The system is currently under design; However, documentation, modeling, and requirements traceability provide a solid foundation for future implementation.

References

Agarwal, V. (2025), 'Designing resilient distributed systems: Fault tolerance strategies and insights', *International Journal of Research in Computer Applications and Information Technology (IJRCAIT)* **8**(1), 1102–1113.

Alrumiah, S. S. and Hadwan, M. (2021), 'Implementing big data analytics in e-commerce: Vendor and customer view', *IEEE Access* **9**, 37281–37286.

Yaqoob, I., Hashem, I. A. T., Gani, A., Mokhtar, S., Ahmed, E., Anuar, N. B. and Vasilakos, A. V. (2016), 'Big data: From beginning to future', *International Journal of Information Management* **36**(6, Part B), 1231–1247.

Zaharia, M., Ghodsi, A., Xin, R., Stoica, I. and Armbrust, M. (2021), 'Lakehouse: A new generation of open platforms that unify data warehousing and advanced analytics', *CIDR Conference* .