



Database Systems

ACCA7048

Ridwa Nul Hoque | MSc Software Engineering (Web Development) | 17/01/2025

Abstract

This report discusses the differences between SQL and NoSQL databases, focusing on their suitability for various applications. It provides recommendations for Pronto Parcels, helping them choose the right database system for their needs. For Eco Bike Hubs, the report identifies key database operations, suggests performance improvements using indexing, and explains how transactions can address operational issues like data consistency and preventing booking conflicts.

Table of Contents

Abstract	1
SQL vs. NoSQL Databases	1
Overview of SQL Databases	1
Overview of NoSQL Databases	1
Key Differences	2
Recommendations for Pronto Parcels	2
PERFORMANCE TUNING FOR ECO BIKE HUBS	2
Key Transactions and Queries	2
CONCLUSION	3
REFERENCES	3

SQL vs. NoSQL Databases

OVERVIEW OF SQL DATABASES

SQL databases are relational, meaning they store data in structured tables that are connected through relationships. They are best suited for systems where:

- Data needs to be consistent and follow strict integrity rules.
- Relationships between data (like orders and customers) are important.
- Operations require ACID properties (Atomicity, Consistency, Isolation, Durability).

Some common examples of SQL databases include PostgreSQL, MySQL, and Microsoft SQL Server.

OVERVIEW OF NOSQL DATABASES

NoSQL databases, on the other hand, are designed to handle large amounts of data with flexibility. They don't use structured tables and are often document-based, key-value stores, or graph databases. These are ideal for:

- Applications needing scalability and high performance.
- Scenarios involving unstructured or semi-structured data.
- Systems with rapidly changing requirements or unpredictable schema.

Examples include MongoDB, Cassandra, and DynamoDB.

KEY DIFFERENCES

<i>Feature</i>	<i>SQL Databases</i>	<i>NoSQL Databases</i>
Data Structure	Relational (tables)	Flexible (documents, key-value)
Scalability	Vertical scaling	Horizontal scaling
Schema	Fixed schema	Schema-less
Query Language	SQL	Varies (e.g., MongoDB uses JSON-like queries)
Use Cases	Financial systems, e-commerce	Real-time analytics, IoT

Recommendations for Pronto Parcels

For Pronto Parcels, an SQL database is ideal if the system needs:

- Strong data consistency and complex relationships (e.g., tracking parcel movements and customer orders).
- Transactional integrity for financial transactions.

However, a NoSQL database could be better if the system involves:

- Real-time tracking of delivery data.
- Handling large volumes of dynamic or rapidly changing data.

PERFORMANCE TUNING FOR ECO BIKE HUBS

KEY TRANSACTIONS AND QUERIES

The following operations are critical for Eco Bike Hubs and need to be improved:

1. Rental history by customer: Quickly retrieving all rentals for a specific customer.
2. Repair lookups: Efficiently finding repairs based on bicycle or customer details.
3. Checking bikes availability: Ensuring no overlapping rentals when booking new ones.

RECOMMENDED INDEXES AND JUSTIFICATIONS

To improve the performance of these queries, the following indexes are suggested:

It will speed up queries retrieving customer rental history.

```
Query Query History ✓
1 SELECT * FROM Rentals WHERE CustomerID = 1;
2
```

This facilitates checks for overlapping bookings.

```
Query Query History
1 SELECT * FROM Rentals WHERE BikeID = 3
2 AND RentalStartDate < '2025-01-20'
3 AND RentalEndDate > '2025-01-15';
4
```

CONCLUSION

This report studies SQL and NoSQL databases, offering recommendations for Pronto Parcels based on their specific needs. For Eco Bike Hubs, the report identified key performance enhancements, such as using indexes and implementing transactions to handle data consistency and booking conflicts.

REFERENCES

- [1] PostgreSQL Documentation, "PostgreSQL: The world's most advanced open source relational database," [Online]. Available: <https://www.postgresql.org/docs/>.
- [2] MongoDB Documentation, "MongoDB Manual," [Online]. Available: <https://www.mongodb.com/docs/>.
- [3] E. F. Codd, "A Relational Model of Data for Large Shared Data Banks," *Communications of the ACM*, vol. 13, no. 6, pp. 377–387, Jun. 1970.
- [4] IBM Cloud, "SQL vs. NoSQL Databases: What's the Difference?" [Online]. Available: <https://www.ibm.com/cloud/learn/sql-vs-nosql>.