

## Database Research: SQL Server vs MySQL vs MongoDB

### Comparison Chart:

Criteria	SQL Server	MySQL	MongoDB
<b>Performance</b>	<ul style="list-style-type: none"><li>- High performance for complex queries and large datasets</li><li>- Optimized for transactional operations (ACID compliance)</li></ul>	<ul style="list-style-type: none"><li>- High read/write performance for structured data</li><li>- Optimized for both transactional and read-heavy applications</li></ul>	<ul style="list-style-type: none"><li>- Excellent for high-velocity data</li><li>- Optimized for unstructured or semi-structured data with flexible schema</li></ul>
<b>Scalability</b>	<ul style="list-style-type: none"><li>- Vertical scaling (can scale on more powerful hardware)</li><li>- Limited horizontal scaling</li></ul>	<ul style="list-style-type: none"><li>- Supports both vertical and horizontal scaling</li><li>- Widely used in distributed systems</li></ul>	<ul style="list-style-type: none"><li>- Excellent horizontal scalability across clusters</li><li>- Ideal for large-scale distributed databases</li></ul>
<b>Security</b>	<ul style="list-style-type: none"><li>- Robust security features (encryption, role-based access, auditing)</li><li>- Meets enterprise compliance requirements</li></ul>	<ul style="list-style-type: none"><li>- Provides good security options (encryption, SSL, role-based access)</li><li>- Supports security best practices for transactional systems</li></ul>	<ul style="list-style-type: none"><li>- Security available (encryption, SSL), but ACID compliance is weaker</li><li>- Data consistency can be a challenge in some cases</li></ul>
<b>Compatibility</b>	<ul style="list-style-type: none"><li>- Highly compatible with Microsoft ecosystem (Windows, Azure, .NET)</li><li>- Ideal for enterprise applications</li></ul>	<ul style="list-style-type: none"><li>- Widely compatible with various programming languages and technologies (Spring Boot, Node.js, etc.)</li><li>- Flexible for multiple operating systems (Windows, Linux)</li></ul>	<ul style="list-style-type: none"><li>- Best suited for NoSQL-based applications</li><li>- High compatibility with JavaScript-based frameworks and cloud platforms</li></ul>
<b>Cost</b>	<ul style="list-style-type: none"><li>- Licensing cost can be high, especially for enterprise editions</li></ul>	<ul style="list-style-type: none"><li>- Open-source version is free</li><li>- Offers a cost-effective option with paid support</li></ul>	<ul style="list-style-type: none"><li>- Open-source, free to use with paid support options</li><li>- Can be cost-effective for large-scale deployments</li></ul>

## Summary of Findings:

### 1. SQL Server:

- **Strengths:** SQL Server is an enterprise-level database with robust performance for handling complex queries and transactions. It is ACID-compliant, making it a strong choice for mission-critical applications that require high consistency and reliability. It also provides powerful security and compliance features.
- **Weaknesses:** SQL Server can be expensive due to licensing fees and is less flexible when it comes to horizontal scaling. It is best suited for applications already in the Microsoft ecosystem.

### 2. MySQL:

- **Strengths:** MySQL offers a good balance of performance, scalability, and security. It's widely adopted due to its open-source nature and extensive support for multiple platforms. MySQL supports ACID transactions and provides good read/write performance for structured data. It is also highly compatible with many web development frameworks, making it versatile for various applications.
- **Weaknesses:** Horizontal scaling in MySQL requires additional tools (like MySQL Cluster), and while it handles large datasets well, it may not be as scalable as NoSQL databases for massive unstructured data.

### 3. MongoDB:

- **Strengths:** MongoDB is a NoSQL database optimized for high-velocity, large-scale data storage. It excels in horizontal scaling and is highly flexible, making it ideal for applications that handle unstructured or semi-structured data. Its flexible schema design is excellent for projects that need rapid development and schema evolution.
- **Weaknesses:** MongoDB sacrifices some ACID properties for scalability and performance, which might not suit applications requiring high data consistency. It is also not designed for complex relational queries like traditional SQL databases.

## Conclusion:

After evaluating the three databases based on performance, scalability, security, and compatibility, **MySQL** has been chosen for the project for the following reasons:

- **Cost-Effectiveness:** MySQL is open-source, which is a cost-effective solution compared to SQL Server's enterprise-level licensing fees.
- **Scalability:** MySQL supports both vertical and horizontal scaling, making it suitable for the project's current requirements, with room to scale as needed.
- **Performance:** MySQL performs well with structured data and provides ACID compliance, making it reliable for transactional operations required by the restaurant management system.
- **Compatibility:** MySQL integrates seamlessly with the existing technology stack (Spring Boot, React, and AWS), making it easy to work with in terms of setup and maintenance.