

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



# Elasticsearch vs. Traditional Databases: Diving into Elastic search's Strengths



Rajeev Kumar

[Follow](#)

3 min read · Aug 30, 2023



When you're selecting a database or search engine for a project, understanding each option's nuances is crucial. Today, we're diving deep into Elasticsearch's strengths and exploring how it compares to traditional SQL and NoSQL databases.

## 1. Introduction to Elasticsearch

Elasticsearch, underpinned by the powerful Apache Lucene library, is a distributed search and analytics engine. It's known for its speed, scalability, and ability to index vast amounts of data rapidly. Unlike many traditional databases, Elasticsearch is tailored for search-centric applications, offering features that many databases don't natively support.



Credits: nextbrick.com

## 2. Elastic search's Strengths

### a. Full-text Search Capabilities

- **Inverted Indexing:** At its core, Elasticsearch uses an inverted index, a data structure that lists every unique word and its corresponding locations in the data. This structure is optimized for speed, enabling rapid text searches across massive datasets.
- **Advanced Text Processing:** Out of the box, Elasticsearch provides capabilities like tokenization (breaking text into individual words or terms), stemming (reducing words to their root form), and handling synonyms. These processes enhance search relevancy and precision.
- **Relevance Scoring and Ranking:** When you search in Elasticsearch, it doesn't just find matches — it ranks them based on relevance using a variety of algorithms, ensuring the most pertinent results are returned

first.

## **b. Flexibility with Data**

- **JSON-Native Structure:** Elasticsearch treats data as JSON documents. This format is not only ubiquitous in modern web applications but also allows for data to be structured hierarchically, enabling more complex queries.
- **Dynamic Mapping:** Unlike some databases that require a fixed schema, Elasticsearch can automatically detect and index the data types of fields in your documents. This flexibility is beneficial for evolving datasets.

## **c. Bulk Indexing**

- **Efficient Data Ingestion:** Elasticsearch's bulk API allows for multiple indexing, updating, or deleting operations in a single request. This streamlined approach ensures high-speed data ingestion, especially when dealing with vast amounts of information.
- **Parallel Processing:** Elasticsearch is designed to handle simultaneous indexing operations across its distributed nodes. This concurrent processing ensures that high volumes of data are indexed rapidly.

## **d. Distributed by Design**

- **Sharding and Replication:** Data in Elasticsearch is inherently divided into "shards." These shards can be replicated across nodes, providing both scalability (by adding more shards) and resilience (through replicas). As your data grows, Elasticsearch grows with you.

# Access to all of it.

Become a Medium member  
for unlimited reading.

Upgrade now

- **Horizontal Scalability:** Need to handle more data? Simply add more nodes to your Elasticsearch cluster. The system automatically distributes your data and query load, ensuring optimal performance.
- **Fault Tolerance:** In the event of a node failure, the presence of replica shards ensures that your data remains available and your search operations continue without disruption.

## e. Real-time Indexing

- **Near Instant Data Availability:** Once ingested, data in Elasticsearch is made available for search operations almost immediately. This real-time indexing capability is thanks to its optimized refresh intervals, ensuring your applications always have access to the latest data.
- **Optimized for High Throughput:** Beyond just real-time indexing, Elasticsearch is designed to handle a continuous stream of data updates, making it particularly effective for time-sensitive applications such as log monitoring or live data analytics.

## 3. Where Traditional Databases Might Have the Edge

### a. ACID Transactions

- SQL databases often prioritize strong ACID guarantees, making them better suited for applications requiring strict data integrity and

consistency.

## **b. Complex Relationships**

- SQL databases are designed around data normalization and relationships. They excel at handling complex joins and relational data modelling.

## **c. General-Purpose Use Cases**

- While Elasticsearch excels at search and analytics, SQL databases are more versatile and suitable for a wide variety of applications.

## **4. Scenarios Best Suited for Elasticsearch**

- **Logging and Monitoring:** Due to its ability to handle high volumes of data and make it searchable in near real-time.
- **Full-Text Search Applications:** Such as e-commerce platforms where a combination of search, filtering, and ranking is essential.
- **Analytics and Visualization:** Tools like Kibana can transform Elasticsearch into a powerful platform for data visualization.

## **5. Conclusion**

In the vast world of databases and search engines, each tool has its unique strengths. While Elasticsearch is undeniably potent for search-centric and data-heavy applications, it's essential to assess the needs of your specific project. By understanding the capabilities and trade-offs of each option, you can ensure that you're leveraging the right tool for your unique challenges.

Elasticsearch

System Design Concepts

Software Architecture



**Written by Rajeev Kumar**

71 followers · 205 following

Follow

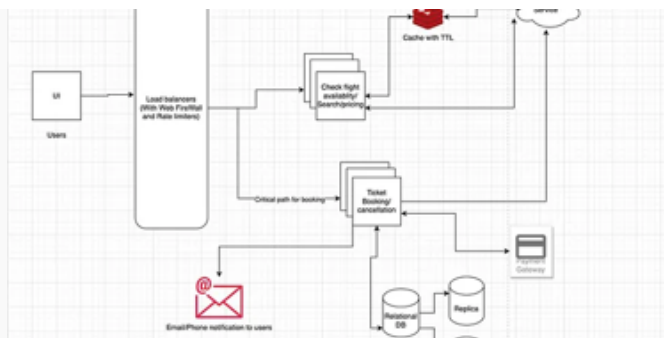
**No responses yet**




Ishu

What are your thoughts?

**More from Rajeev Kumar**



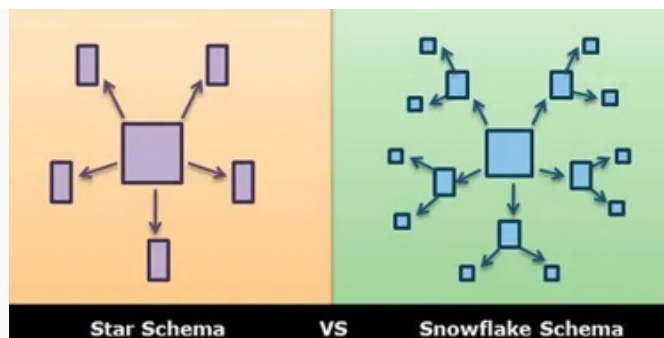
 Rajeev Kumar


## Make my trip System design.

Functional Requirements:

Dec 27, 2023

 9  1



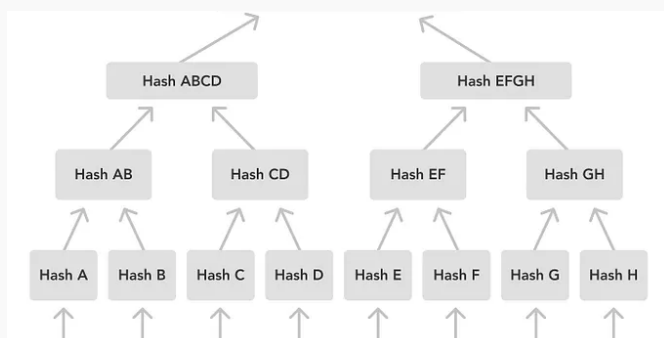
 Rajeev Kumar


## Understanding Data Modeling: From Facts to Galaxy Schemas

Data modelling is an essential practice in today's data-driven world. From...

Aug 29, 2023

 3



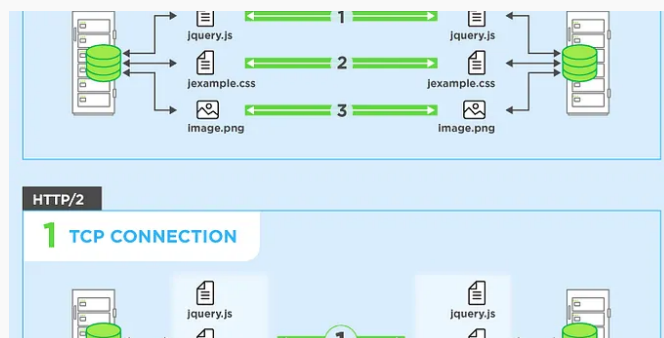
 Rajeev Kumar


## Understanding Merkle Trees: The Backbone of Data Verification

Amid the numerous structures in computer science, the Merkle tree holds a special...

Aug 14, 2023

 22



 Rajeev Kumar

## From HTTP/1.1 to HTTP/2: A New Era of Web Performance

Introduction

Aug 19, 2023

 3



See all from Rajeev Kumar

## Recommended from Medium



Aditya Bhuyan

### **Bridging Elasticsearch and PostgreSQL: A Deep Dive into...**

Introduction



Jul 16



8



Süleyman Kütükoğlu

### **How to Migrate from Ingress NGINX to Gateway API (Step-by...**

Ingress-NGINX will be fully retired in March 2026 and will no longer receive bug fixes o...




Nov 14



6






 Arvind Kumar

## Spring Framework 7 Is Here: Why This Release Matters More Than...

A practical, developer-first guide to the biggest changes—and what you should...

★ 6d ago 🤝 4




 Swapnil agrawal

## Availability in System Design

Availability is the proportion of time a system is able to serve incoming requests...

★ Nov 3




 In Stackademic by Dewan Mahmud Raihan

## Supercharge Spring Boot REST APIs with Non-Blocking...

Build Production-Ready Observability with Async Logging, Spring AOP, and ELK Stac...

★ Jul 28 🤝 50



 In InfraMind by Faris Shabibi

## Networking For Devops : Part 1

If you've ever scratched your head hearing terms like IP address, TCP/IP, or subnettin...

★ Jun 18 🤝 5



See more recommendations

[Help](#) [Status](#) [About](#) [Careers](#) [Press](#) [Blog](#) [Privacy](#) [Rules](#) [Terms](#) [Text to speech](#)