Module 1 –Introduction to Mongo DB –Architecture and Installation-Assignment

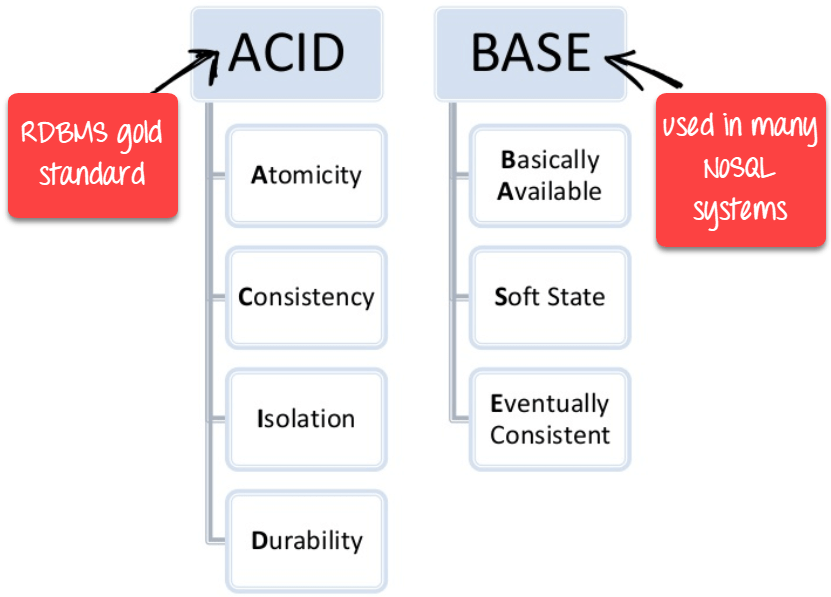
1. **List the business requirements-**

* Enable business innovation
* Reduces cost
* Aaccelerates time to market
* Transforms customer experience
* Ehances efficiency.

1. **Compare different database categories**

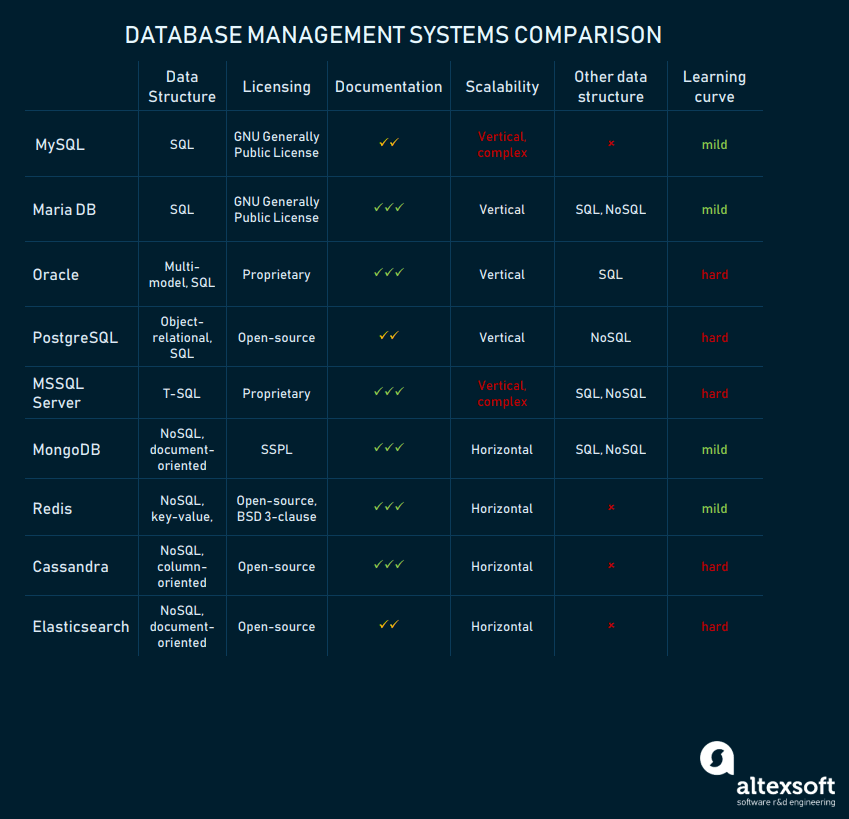
Below is the main difference between NoSQL and SQL:

|  |  |  |
| --- | --- | --- |
| Parameter | SQL | NOSQL |
| Definition | SQL databases are primarily called RDBMS or Relational Databases | NoSQL databases are primarily called as Non-relational or distributed database |
| Design for | Traditional RDBMS uses SQL syntax and queries to analyze and get the data for further insights. They are used for OLAP systems. | NoSQL database system consists of various kind of database technologies. These databases were developed in response to the demands presented for the development of the modern application. |
| Query Language | Structured query language (SQL) | No declarative query language |
| Type | SQL databases are table based databases | NoSQL databases can be document based, key-value pairs, graph databases |
| Schema | SQL databases have a predefined schema | NoSQL databases use dynamic schema for unstructured data. |
| Ability to scale | SQL databases are vertically scalable | NoSQL databases are horizontally scalable |
| Examples | Oracle, Postgres, and MS-SQL. | [MongoDB](https://www.guru99.com/mongodb-tutorials.html), Redis, Neo4j, Cassandra, Hbase. |
| Best suited for | An ideal choice for the complex query intensive environment. | It is not good fit complex queries. |
| Hierarchical data storage | SQL databases are not suitable for hierarchical data storage. | More suitable for the hierarchical data store as it supports key-value pair method. |
| Variations | One type with minor variations. | Many different types which include key-value stores, document databases, and graph databases. |
| Development Year | It was developed in the 1970s to deal with issues with flat file storage | Developed in the late 2000s to overcome issues and limitations of SQL databases. |
| Open-source | A mix of open-source like Postgres & MySQL, and commercial like Oracle Database. | Open-source |
| Consistency | It should be configured for strong consistency. | It depends on DBMS as some offers strong consistency like MongoDB, whereas others offer only offers eventual consistency, like [Cassandra](https://www.guru99.com/cassandra-tutorial.html). |
| Best Used for | RDBMS database is the right option for solving ACID problems. | NoSQL is a best used for solving data availability problems |
| Importance | It should be used when data validity is super important | Use when it's more important to have fast data than correct data |
| Best option | When you need to support dynamic queries | Use when you need to scale based on changing requirements |
| Hardware | Specialized DB hardware (Oracle Exadata, etc.) | Commodity hardware |
| Network | Highly available network (Infiniband, Fabric Path, etc.) | Commodity network (Ethernet, etc.) |
| Storage Type | Highly Available Storage (SAN, RAID, etc.) | Commodity drives storage (standard HDDs, JBOD) |
| Best features | Cross-platform support, Secure and free | Easy to use, High performance, and Flexible tool. |
| Top Companies Using | Hootsuite, CircleCI, Gauges | Airbnb, Uber, Kickstarter |
| Average salary | The average salary for any professional SQL Developer is $84,328 per year in the U.S.A. | The average salary for "NoSQL developer" ranges from approximately $72,174 per year |
| ACID vs. BASE Model | [ACID](https://www.guru99.com/dbms-transaction-management.html)( Atomicity, Consistency, Isolation, and Durability) is a standard for RDBMS | Base ( Basically Available, Soft state, Eventually Consistent) is a model of many NoSQL systems |

[](https://www.guru99.com/images/1/101818_0550_Differenceb2.png)

Difference between ACID vs BASE in DBMS

**• Select a database from the category which caters to the business needs**



## MySQL

This is one of the most popular relational database systems. Originally an open-source solution, MySQL now is owned by Oracle Corporation. Today, MySQL is a pillar of LAMP application software. That means it’s a part of Linux, Apache, MySQL, and Perl/PHP/Python stack. Having C and C++ under the hood, MySQL works well with such system platforms as Windows, Linux, MacOS, IRIX, and others.

### **Pros of MySQL**

**Free installation.**The community edition of MySQL is free to download. With a basic set of tools for individual use, [MySQL community edition](https://dev.mysql.com/downloads/) is a good option to begin with. Of course, there are other, prepaid options for [Enterprise](https://support.oracle.com/epmos/faces/MosIndex.jspx?_afrLoop=255034323681101&_afrWindowMode=0&_adf.ctrl-state=1bvk9vhmkx_9) or [Cluster](https://support.oracle.com/) purposes with richer functionality. Nevertheless, if your company is too small to pay for one of them, the free-to-download model is the most suitable for a fresh start.

**Simple syntax and mild complexity.**MySQL’s structure and style are very plain. Developers even consider MySQL a database with a human-like language. As MySQL is often used in tandem with PHP programming language. Because they share a gentle learning curve, you won’t need to hire a skilled developer to manage your database. Also, MySQL is easy to use. For instance, most of the tasks can be executed right in the command line, reducing development steps.

**Cloud-compatible.**Business-oriented by nature and originally developed for the web, MySQL is supported by the most popular cloud providers. It’s available on such leading platforms as Amazon, Microsoft, and others. This makes MySQL even more attractive and gives businesses using it room for growth.

### **Cons of MySQL**

**Scalability challenges.**MySQL was not built with scalability in mind, which is inherent in its code. In theory, you can scale MySQL, but it will need more engineering effort as compared to any of the NoSQL databases. So, if you expect one day your database will increase substantially, keep this limitation in mind or choose another DBMS option.

**Partial open source.**Although MySQL has the open-source part, it’s mostly under Oracle’s license. This limits the MySQL community in terms of improving the DBMS. Why do you care? Because when you have completely open-source support, you expect many problem-specific implementations and community assistance. This is not the case when the software belongs to corporate owners and you’ll have to pay for support.

**Limited compliance with SQL standards.**Structured Query Language has specific standards. MySQL doesn’t completely follow them, i.e. MySQL provides no support for some standard SQL features. On the other hand, MySQL has some extensions and distinct features that don’t match the Structured Query Language standards. It’s not a big deal for small web applications. The issues may appear when you have to shift to other databases, which is likely to happen when your business starts growing.

### **Use cases**

**Small web-based solutions.**MySQL database system is the best option when you’re designing a small, web-based solution with a small volume of data. For example, when building a local eCommerce store, MySQL may come in handy.

**OLAP/OLTP systems.** This is one of the best use cases for a MySQL database, as OLAP/OLTP don’t require complex queries and large volumes of data. Also, consider applying MySQL for the same reason if you’re building a [business intelligence tool](https://www.altexsoft.com/blog/business/complete-guide-to-business-intelligence-and-analytics-strategy-steps-processes-and-tools/).

## Oracle

Oracle is a relational database management system created and run by the Oracle Corporation. Currently, it supports multiple data models like document, graph, relational, and key-value within the single database. In its latest releases, it refocused on cloud computing. Oracle database engine licensing is fully proprietary, with both free and paid options available.

### **Pros of Oracle**

**Innovations for daily workflow.**With [Oracle 12c](https://www.oracle.com/database/12c-database/) as hybrid cloud software, innovative cloud computing technologies show up daily. At the same time, it keeps focusing on information security. Besides active data guard, partitioning, improved backup, and recovery, Oracle suggests parallel upgrading to reduce downtime during database upgrades.

**Strong tech support and documentation.**Oracle ensures decent customer support and provides comprehensive tech documentation across multiple resources. So, you’ll likely find solutions to any issues that appear. You may also expect some community support.

**Large capacity.**Oracle’s multi-model solution allows for accommodating and processing a vast amount of data. Thanks to the recently released multi-tenancy feature, the database architecture now simplifies packing many databases and manage them smoothly. In combination with in-memory data processing capabilities, it creates a strong engine for synchronous data processing.

### **Cons of Oracle**

**High cost.**Though Oracle 12c RDBMS has free editions, they are very limited in terms of functionality. Standard Edition, which doesn’t include all available features, costs $17,500 per unit. The Enterprise Edition is over $47,000 per unit.

**Resource-consuming.**Oracle database needs powerful infrastructure. Not only does installation require a lot of disk space, you’ll also have to consider constant hardware updates if you deploy it on premises.

**Hard learning curve.**Oracle database is not a system to start using right away. It’s better to have certified Oracle DB engineers to run it. Oracle’s documentation, while covering many issues, can sometimes be overwhelming and even confusing. So, to install and run an Oracle database, you’ll have to consider hiring dedicated experts.

### **Use cases**

Given all those perks and pitfalls, you can consider Oracle RDMS as a reasonable solution for online OLTP, data warehousing, and even mixed (OLTP and DW) database application. If you have a billion records to hold and manage – and sufficient budget to support it – Oracle hybrid cloud software is a good option to choose.

## PostgreSQL

This database management system shares its popularity with MySQL. This is an object-relational DBMS where user-defined objects and table approach are combined to build more complex data structures. Besides that, PostgreSQL has a lot of similarities with MySQL. It’s aimed at strengthening the standards of compliance and extensibility. Consequently, it can process any workload, for both single-machine products and complex applications. Owned and developed by PostgreSQL Global Development Group, it still remains a completely open source. This DBMS is available for use with such platform systems as Microsoft, iOS, Android, and many more.

### **Pros of Postgre**

**Scalable.**Vertical scalability is a hallmark of PostgreSQL, unlike MySQL DBMS. Considering that almost any custom software solution tends to grow, resulting in database extension, this particular option certainly supports business growth and development.

**Support for custom data types.** PostgreSQL natively supports a large number of data types by default, such as JSON, XML, H-Store, and others. PostgreSQL takes advantage of it, being one of the few relational databases with strong support for NoSQL features. Additionally, it allows users to define their own data types. As your [software business model](https://www.altexsoft.com/blog/business/software-business-models-examples-revenue-streams-and-characteristics-for-products-services-and-platforms/) may need different types of databases throughout its existence for better performance or application comprehensiveness, this option brings improved flexibility to the table.

**Easily-integrated third-party tools.**PostgreSQL database management system has the strong support of [additional tools](https://wiki.postgresql.org/wiki/Community_Guide_to_PostgreSQL_GUI_Tools), both free and commercial. The scope of these includes extensions to improve many aspects. For example, [ClusterControl](https://severalnines.com/product/clustercontrol) provides impressive assistance at managing, monitoring, and scaling SQL and NoSQL open source databases. To make data comparison and synchronization more effective, consider using DB Data Difftective. In case you’re going to scale up your data to heavy workloads, [pgBackRest](https://pgbackrest.org/) backup and restore system will be a nice option to choose.

**Open-source and community-driven.** Postgres is completely open-source and supported by its community, which strengthens it as a complete ecosystem. Additionally, developers can always expect free and prompt community assistance.

### **Cons of Postgre**

**Inconsistent documentation.**While PostgreSQL has a large community and provides strong support for its participants, the documentation still lacks consistency and completeness. As the PostgreSQL community is rather distributed, the documentation doesn’t follow equal standards for all Postgre features.

**Lack of reporting and auditing instruments.** A significant shortcoming of PostgreSQL is the absence of revising tools that would show the current condition of a database. You have to continuously check if something goes wrong. There’s always a risk that DB engineers will notice a failure too late.

### **Use cases**

Due to complicated queries and a wide choice of custom interfaces accomplished with predefined functions, PostgreSQL is a perfect match for data analysis and warehousing. If you are building a database automation tool, PostgreSQL is the best fit for it due to its strong analytical capabilities, ACID-compliance, and powerful SQL engine. All in one, it significantly accelerates the processing of vast amounts of data. This DBMS is popular with financial institutions and telecommunication systems.

## MSSQL

As a completely commercial tool, Microsoft SQL Server is one of the most popular relational DBMS, in addition to MySQL, PostgreSQL, and Oracle. It copes well with effective storing, changing, and managing relational data. To interact with SQL Server databases, DB engineers usually utilize the [Transact-SQL](https://www.tutorialspoint.com/t_sql/) (T-SQL) language, which is an extension of the SQL standard.

### **Pros of MSSQL**

**Variety of versions.**Microsoft SQL Server provides a [wide choice of different options](https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2017?view=sql-server-2017#includessnoversionincludesssnoversion-mdmd-editions) with diverse functionalities. For instance, the Express edition with a free database offers entry-level tooling, the perfect match for learning and building desktop or small server data-driven applications. The Developers option allows for building and testing applications including some enterprise functionalities, but without a production server license. For bigger projects, there are also Web, Standard, and Enterprise editions, with a varying extent of administrative capabilities and service levels.

**End-to-end business data solution.**With a focus on mostly commercial solutions, MSSQL provides a lot of business value-added features. The optional selection of components allows building ETL solutions, forming a knowledge base, and implementing data clearance. Also, it provides tools for overall data administration, online analytical processing, and data mining, additionally providing options for report and visualization generation.

**Rich documentation and community assistance.**With Microsoft SQL Server aimed at comprehensive database maintenance, the full online documentation also reflects this concept. The consequently structured guidelines, numerous whitepapers, and demos give a full picture on the MSSQL data system. Also, Microsoft Premier provides access to dedicated Microsoft community support, which is an advantage when a DB engineer needs assistance.

**Cloud database support.** Being a part of the consistent Microsoft ecosystem, MSSQL can be integrated with Microsoft cloud, Azure SQL Database, or SQL Server on Azure Virtual Machines. The solutions allow shifting database administration to the cloud if your business software database becomes really overwhelming and hard to administer.

### **Cons of MSSQL**

**Cost-consuming**. Being mostly used at enterprise scale, MSSQL Server remains one of the most expensive solutions. Speaking of numbers, the Enterprise edition currently costs over $14, 000 per core, sold as 2 core packs.

**Unclear and floating license conditions.**Another issue is the ever-changing licensing process. The pricing strategy itself is hard to understand and the elements included in a particular edition are floating, tending to shift from one to another.

**Complicated tuning process.**For those beginners who have to operate heavy data sets, working with query optimization and performance tuning may be problematic. As the process is not so obvious, it can create substantial bottlenecks early on.

### **Use cases**

MSSQL Server is a reasonable option for companies with other Microsoft product subscriptions. As Microsoft creates a sustainable ecosystem with well-integrated services, the MSSQL here with its access to cloud and powerful data retrieval tools comes in handy.

## MongoDB

A free, open-source, non-relational DBMS, MongoDB also includes a commercial version. Although MongoDB wasn’t initially intended for structured data processing, it can be employed for applications that use both structured and unstructured data. In MongoDB, databases are connected to applications via database drivers. They are widely available within the database management system. Multiple types of data are processed simultaneously and use the internal cache for this purpose.

### **Pros of MongoDB**

**Simple data access, storage, input, and retrieval.**One of the benefits of MongoDB derived from its NoSQL nature is the fast and easy data operation. That is to say, data can be entered, stored, and withdrawn from the database quickly and without any additional confirmation. As any other non-relational database, it places emphasis on RAM usage, so the records can be manipulated really fast and without any consequences to data integrity.

**Easy compatibility with other data models.**MongoDB is easily combined with different database management systems, both SQL and NoSQL types. Besides that, it has pluggable storage engine APIs. To make a long story short, this option allows third parties to build their own data storage engines for MongoDB. From a commercial point of view, it creates extra value for business software.

**Horizontally scalable solution.** Scalability – where data is spread out across a distributed network of manageable servers – is a facet of MongoDB’s fundamental nature. It becomes even more important for enterprises operating big data applications. Additionally, the database can allocate data across a cluster of machines. How can that help you? The data is distributed faster and equally, free of bulkiness. As it leads to faster data processing, the application performance is accelerated too.

### **Cons of MongoDB**

**Extensive memory consumption.**The denormalization process. when previously normalized data in a database is grouped to increase performance, usually results in high memory consumption. Also, this DBMS keeps in memory all key names for each value pair. Beyond that, because there is no support for joins, Mongo databases have data oversupply, resulting in big memory waste and lower application performance.

**Data insecurity.**With a focus on fast data operation, MongoDB, like any other NoSQL DBMS, lacks data security. As user authentication isn’t a default Mongo option, and higher protection is available with a commercial edition only, you can’t consider it totally secure. Additionally, there are constant MongoDB update releases, with no guarantee that all amendments or data changes will work as they did before. Keep in mind that all manipulations should be formed around these updates, being covered with additional tests.

**Complicated process to interpret into other query languages.**As MongoDB wasn’t initially developed to deal with relational data models, the performance may slow down in these cases. Besides, the translation of SQL to MongoDB queries takes additional action to use the engine, which may delay the development and deployment.

### **Use cases**

MongoDB works best in real-time data integration and database scalability. For instance, it’s the right option for product catalogs due to its capacity to stock a multiplicity of objects with various attribute collections. Also, consider here analytic platforms, as MongoDB’s speed provides dynamic performance that can help track the user’s behavior in real time.

## Redis

An open-source, NoSQL, in-memory data structure store, Redis can also be used as a cache. Instead of documents, it uses key-value pairs. Its distinct feature is that there are several options for data structuring, such as lists, sets, and hashes.

Allowing for data replication and supporting transactions, Redis executes commands in a queue instead of setting it one at a time.

### **Pros of Redis**

**Rapid solution.**Due to its replication and transaction features, Redis processes the data really fast. The absence of dependencies and in-memory data store type makes Redis a worthy competitor even among simple SQL alternatives.

**Massive data processing.** From the data perception and refining perspective, Redis can be considered a colossus. It can easily upload up to 1GB of data for one entry. Add built-in data caching and you get a powerhouse data machine.

### **Cons of Redis**

**Requires dataset to fit into memory.**Total reliance and dependency on the application memory is a real drawback. That is to say, your database will crash if its size exceeds the size of available memory.

**No support for query language or joins.**Regarding compatibility with other dataset types, Redis lags behind. Given that at some time your business may need scaling and using other data formats, having rapid entries as a single option leaves this issue open.

### **Use cases**

Redis basically has a few different directions to work with. And the first of them is IoT applications. Here, heavy data from IoT devices can be transferred to Redis to process these records before keeping them in any steady data storage. Also, Redis is a perfect option for [microservice architectures](https://www.altexsoft.com/blog/engineering/using-microservices-for-legacy-system-modernization/) with scalable cloud hosting. As data here doesn’t have to be long term persistent, Redis seems a reasonable decision.

## Cassandra

Cassandra is a decentralized system developed by Apache. Cassandra is a free DBMS whose strength is in its multi-replication and multi-deployment features. These peculiarities allow for numerous query copying and deploying all of them at the same time. Being rapidly scalable, Cassandra allows for managing large data volumes by replicating it into multiple nodes. It eliminates the problem of database crash – if some of the nodes fail at any time, it’s replaced immediately, and the system keeps working as long as at least one single node is safe.

Cassandra uses its own query language, CQL. In its syntax, it’s very similar to SQL but doesn’t apply joins, replacing them with so-called column families. And the second difference is that not all columns in a table are stored for subqueries. Some of them are used as clustering columns, where adjacent data is put next to each other for fast retrieval. Why does that matter? It provides faster querying from massive datasets, accelerating data processing.

### **Pros of Cassandra**

**Data security.**Due to its master node replication feature, Cassandra stays failure tolerant. It means that DB engineers can feel confident about data safety unless master nodes fail all at the same time. As long as it’s extremely unlikely, the database and the application built on it will stay sound and secure.

**Flexibility and on-hand amendments**. Casandra’s simple syntax has the best of SQL and NoSQL. In addition to scalability, it largely contributes to dataset flexibility. Cassandra collects data on the go, and data retrieval shares the same simplicity, despite dataset size. This allows enlarging the database to the fullest extent.

### **Cons of Cassandra**

**Slow reading.**As Cassandra was initially designed for fast writing, its weakness lies in its incapacity for fast reading. One of the reasons for it is that there are no bottlenecks for information sent, so it needs more time to process.

**Requires additional resources.**As Cassandra processes multiple layers of data simultaneously, it demands enough power to do it, resulting in the JVM usage. This means additional investment in both software and hardware. If this is the first time a company faces such a necessity and is not sure about the resources, then maybe it should consider other database systems.

### **Use cases**

Thanks to even data distribution, Cassandra is relevant in applications where large volumes of information are processed. For instance, it’s a great choice for data centers. Also, Cassandra fits well with real-time analytics, as it allows linear scaling and data increase in real time. You may also consider it for applications with constant data streaming like weather apps. Another option is using it as a DBMS for an eCommerce store, as it allows for storing purchase history and other transactions. Add here feasibility to track such data types as order status and packages, and you’ll get the full solution with [eCommerce delivery](https://www.altexsoft.com/blog/business/how-to-integrate-with-ecommerce-delivery-and-shipment-carriers-dhl-fedex-uk-royal-mail-ups-usps-canada-post-amazon/) integration.

## Elasticsearch

Elasticsearch is a NoSQL, document-oriented database management system having a full-text search engine in its heart. Built on the Apache Lucene library, it stores data as a JSON file, supports [RESTful APIs](https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/), and uses a powerful analytical engine for faster data retrieval. Being open-source software, it includes both free and paid editions.

### **Pros of Elasticsearch**

**Scalable architecture.**One of Elasticsearch’s peculiarities is its robust distributed architecture. Its key structure options, such as clustering, indexing, sharding, and many more, provide extensive horizontal scaling, which allows for accommodating terabytes of records with further automation. The architecture’s abstraction levels streamline system management on both individual and aggregate levels.

**Fast data processing.**Due to the distributed data structure and built-in parallelization, the Elasticsearch DB shows excellent performance results. Even when executing a complex data query, it generates lightning search result response. This is partly available due to documents being maintained close to relevant metadata in the index, which makes them fast to find.

### **Cons of Elasticsearch**

**Lack of multi-language support.**When handling request or response data, Elasticsearch DBMS lags behind. Though it’s perfectly combined with Cassandra DB to complement database performance, other languages and formats are not available for it. In these terms, it only supports JSON document format.

**Limited consistent health check tools.**When something goes wrong, as it may at any stage, Elasticsearch can only show status as “yellow” or “red.” Simply put, it has no reporting tools. Though issues are usually like memory threshold or disk capacity, DBA engineers complain about the situation.

### **Use cases**

Due to its NoSQL distributed nature and flexible data models, Elasticsearch is a great tool for eCommerce products with huge databases that tend to use search engines. It’s very helpful when creating or updating a customer’s profile in terms of workload that real-time engagement usually demands.

**•Install the database**

# **Install MongoDB**

Author: MongoDB Documentation Team

This guide describes how to install MongoDB locally. If you would like to use MongoDB in the Cloud using Atlas, our managed database product, see [Get Started with Atlas](https://docs.atlas.mongodb.com/getting-started/).

Time required: 10 minutes

## What You’ll Need

MongoDB supports a variety of 64-bit platforms. Refer to the [Supported Platforms](https://docs.mongodb.com/manual/installation/#mongodb-supported-platforms) table to verify that MongoDB is supported on the platform to which you wish to install it.

## Procedure

### Install MongoDB

* **Windows**
* **macOS**
* **Linux**

Download the binaries from the [MongoDB Download Center](https://www.mongodb.com/download-center#production).

1. Open Windows Explorer/File Explorer.
2. Change the directory path to where you downloaded the MongoDB .msi file. By default, this is %HOMEPATH%\Downloads.
3. Double-click the .msi file.
4. The Windows Installer guides you through the installation process.

If you choose the **Custom** installation option, you may specify an installation directory.

MongoDB does not have any other system dependencies. You can install and run MongoDB from any folder you choose.

**NOTE**

This tutorial assumes that you installed MongoDB in C:\Program Files\MongoDB\Server\4.2\.

### Run MongoDB

* **Windows**
* **macOS**
* **Linux**

**WARNING**

Do not make [mongod.exe](https://docs.mongodb.com/manual/reference/program/mongod.exe/#bin.mongod.exe) visible on public networks without running in “Secure Mode” with the auth setting. MongoDB is designed to be run in trusted environments, and the database does not enable “Secure Mode” by default.

**1 Set up the MongoDB environment**

MongoDB requires a [data directory](https://docs.mongodb.com/manual/reference/glossary/#term-dbpath) to store all data. MongoDB’s default data directory path is the absolute path \data\db on the drive from which you start MongoDB. Create this folder by running the following command in a Command Prompt:

md \data\db

You can specify an alternate path for data files using the [--dbpath](https://docs.mongodb.com/manual/reference/program/mongod/#cmdoption-mongod-dbpath) option to [mongod.exe](https://docs.mongodb.com/manual/reference/program/mongod.exe/#bin.mongod.exe), for example:

"C:\Program Files\MongoDB\Server\4.2\bin\mongod.exe" --dbpath d:\test\mongodb\data

If your path includes spaces, enclose the entire path in double quotes, for example:

"C:\Program Files\MongoDB\Server\4.0\bin\mongod.exe" --dbpath "d:\test\mongo db data"

You may also specify the dbpath in a [configuration file](http://docs.mongodb.com/manual/reference/configuration-options).

**2 Start MongoDB**

To start MongoDB, run [mongod.exe](https://docs.mongodb.com/manual/reference/program/mongod.exe/#bin.mongod.exe). For example, from the **Command Prompt**:

"C:\Program Files\MongoDB\Server\4.0\bin\mongod.exe"

This starts the main MongoDB database process. The waiting for connections message in the console output indicates that the [mongod.exe](https://docs.mongodb.com/manual/reference/program/mongod.exe/#bin.mongod.exe) process is running successfully.

Depending on the security level of your system, Windows may pop up a **Security Alert** dialog box about blocking “some features” of C:\Program Files\MongoDB\Server\4.0\bin\mongod.exe from communicating on networks. All users should select Private Networks, such as my home or work network and click Allow access. For additional information on security and MongoDB, please see the [Security Documentation](http://docs.mongodb.com/manual/security).

**3 Verify that MongoDB has started successfully**

Verify that MongoDB has started successfully by checking the process output for the following line:

[initandlisten] waiting for connections on port 27017

The output should be visible in the terminal or shell window.

You may see non-critical warnings in the process output. As long as you see the log line shown above, you can safely ignore these warnings during your initial evaluation of MongoDB.

**4 Connect to MOngoDB**

To connect to MongoDB through the [~bin.mongo.exe](https://docs.mongodb.com/manual/reference/program/mongo/#bin.mongo) shell, open another **Command Prompt**.

"C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe"

**•Explore the database**

In the market some GUI available to explore database but now, I am only give overview of *MongoDB compass*

The GUI for MongoDB. Visually explore your data. Run ad hoc queries in seconds. Interact with your data with full CRUD functionality. View and optimize your query performance. Available on Linux, Mac, or Windows. Compass empowers you to make smarter decisions about indexing, document validation, and more.

#### **Visualize and explore**

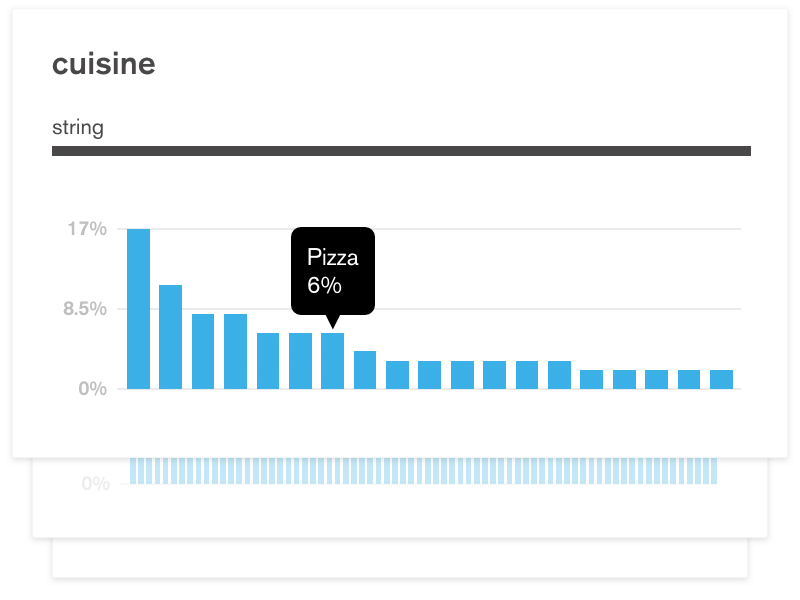
Visualize, understand, and work with your data through an intuitive GUI.

#### **Insert, modify, and delete**

Modify your data with a powerful visual editing tool.

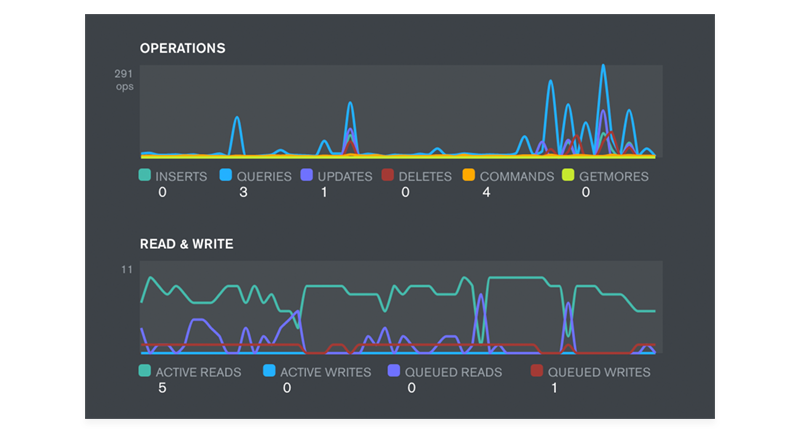
#### **Debug and optimize**

Understand performance issues with visual explain plans, view utilization and manage your indices.



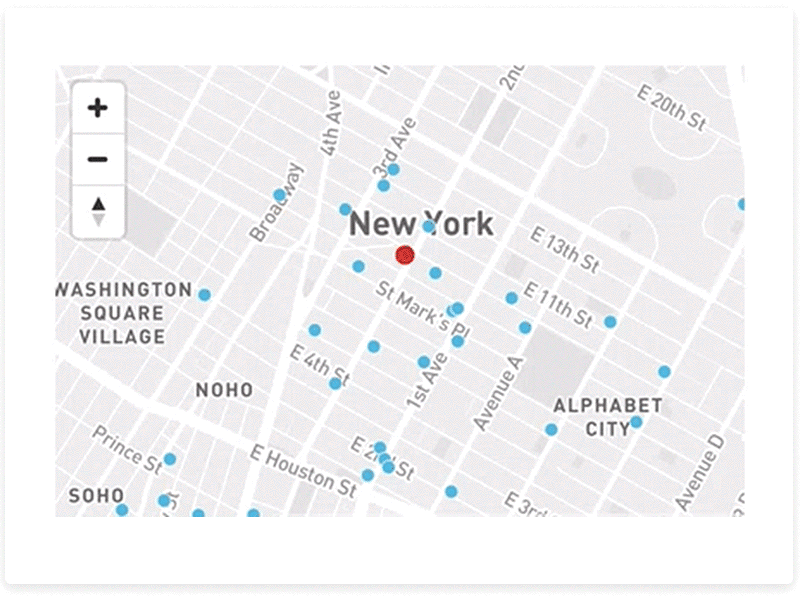
### **Know your data with built-in schema visualization**

MongoDB Compass analyzes your documents and displays rich structures within your collections through an intuitive GUI. It allows you to quickly visualize and explore your schema to understand the frequency, types and ranges of fields in your data set.



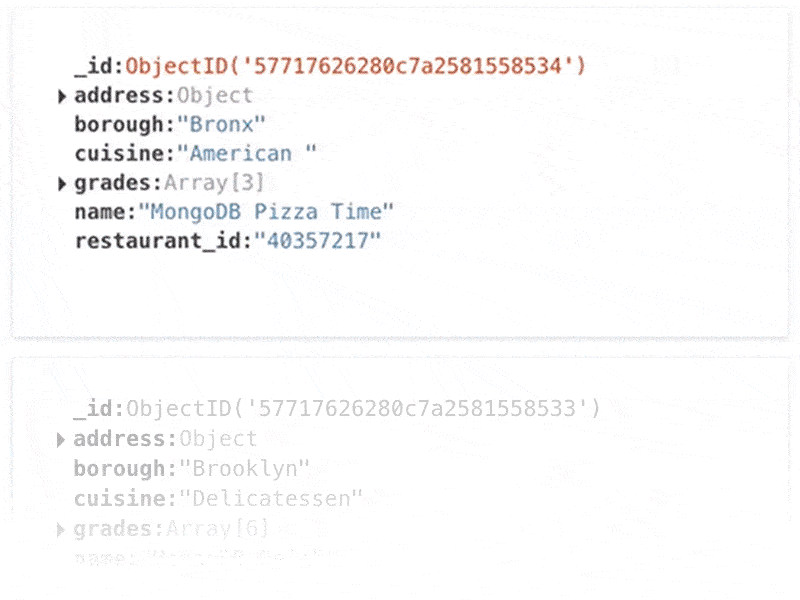
### **Get immediate insight into server status and query performance**

Real-time server statistics let you view key server metrics and database operations. Drill down into database operations easily and understand your most active collections.



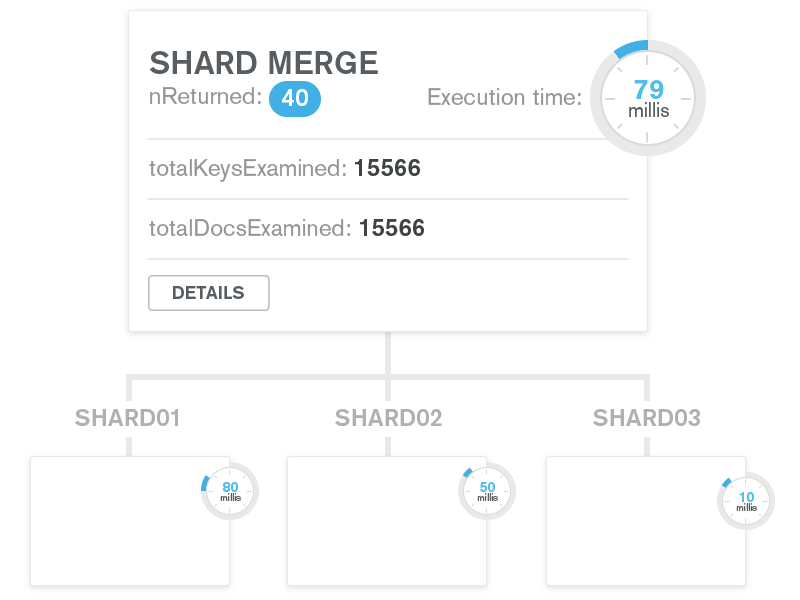
### **Visualize, understand, and work with your geospatial data**

Point and click to construct sophisticated queries, execute them with the push of a button and Compass will display your results both graphically and as sets of JSON documents.



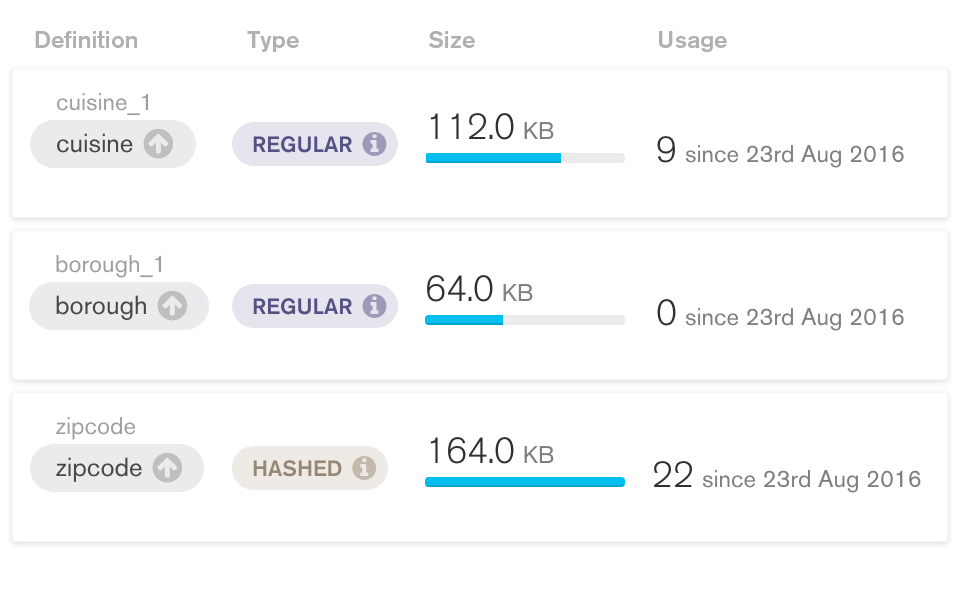
### **A better approach to CRUD makes it easier to interact with your data**

Modify existing documents with greater confidence using the intuitive visual editor, or insert new documents and clone or delete existing ones in just a few clicks.



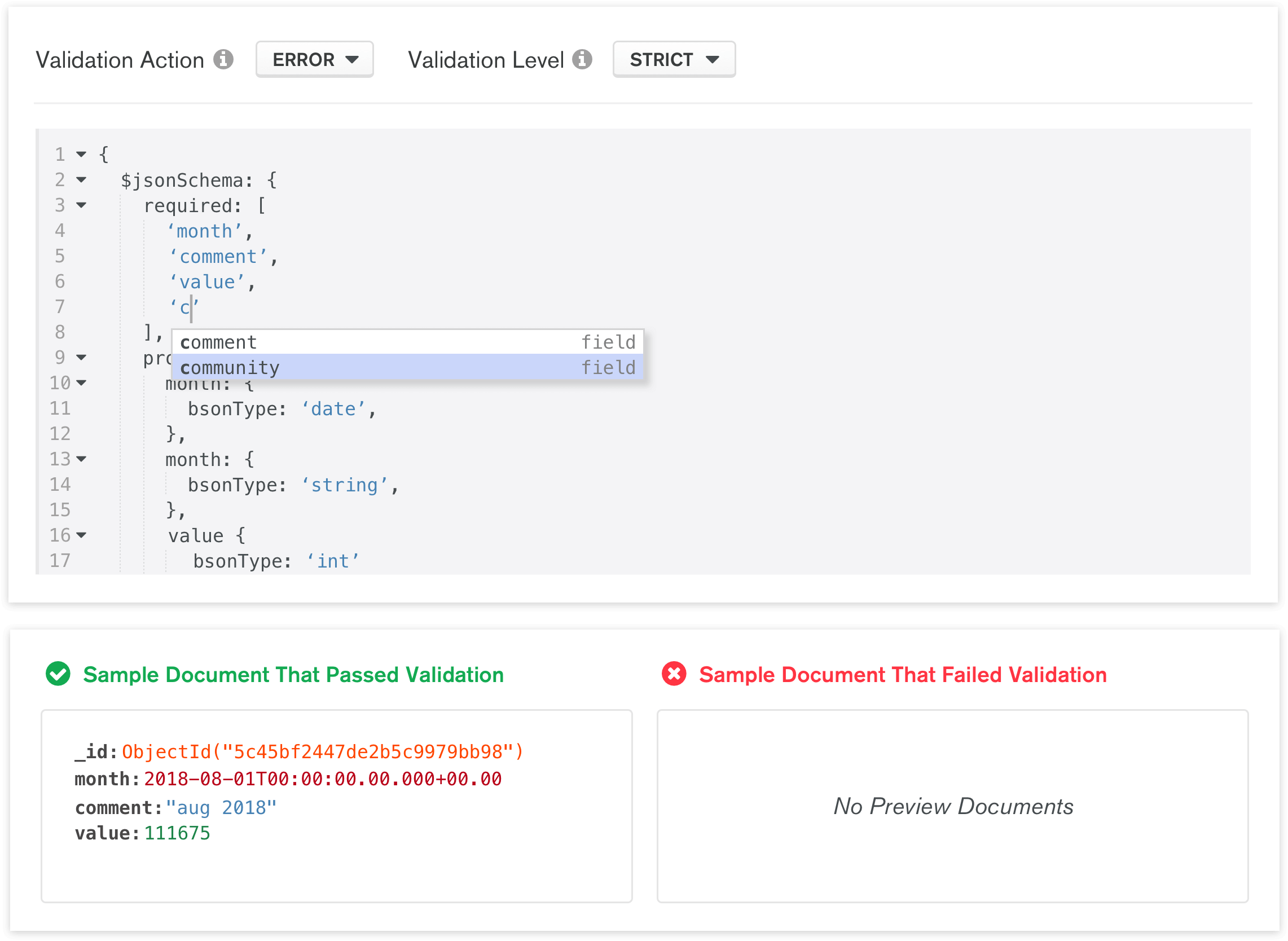
### **Understand performance issues with visual explain plans**

Know how queries are running through an easy-to-understand GUI that helps you identify and resolve performance issues.



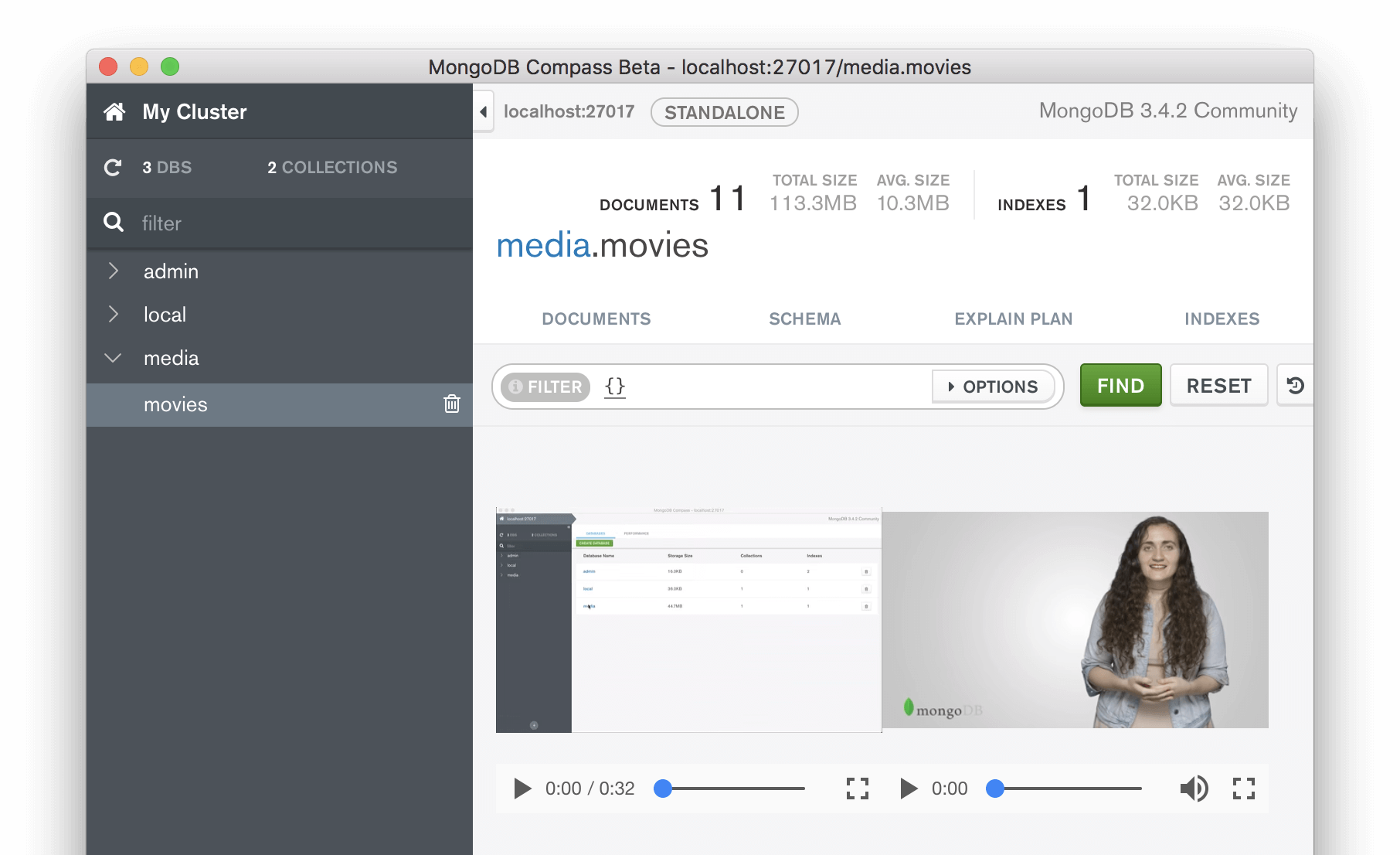
**View utilization and manage your indexes**

Understand the type and size of your indexes, their utilization and special properties. Add and remove indexes at the click of a button.



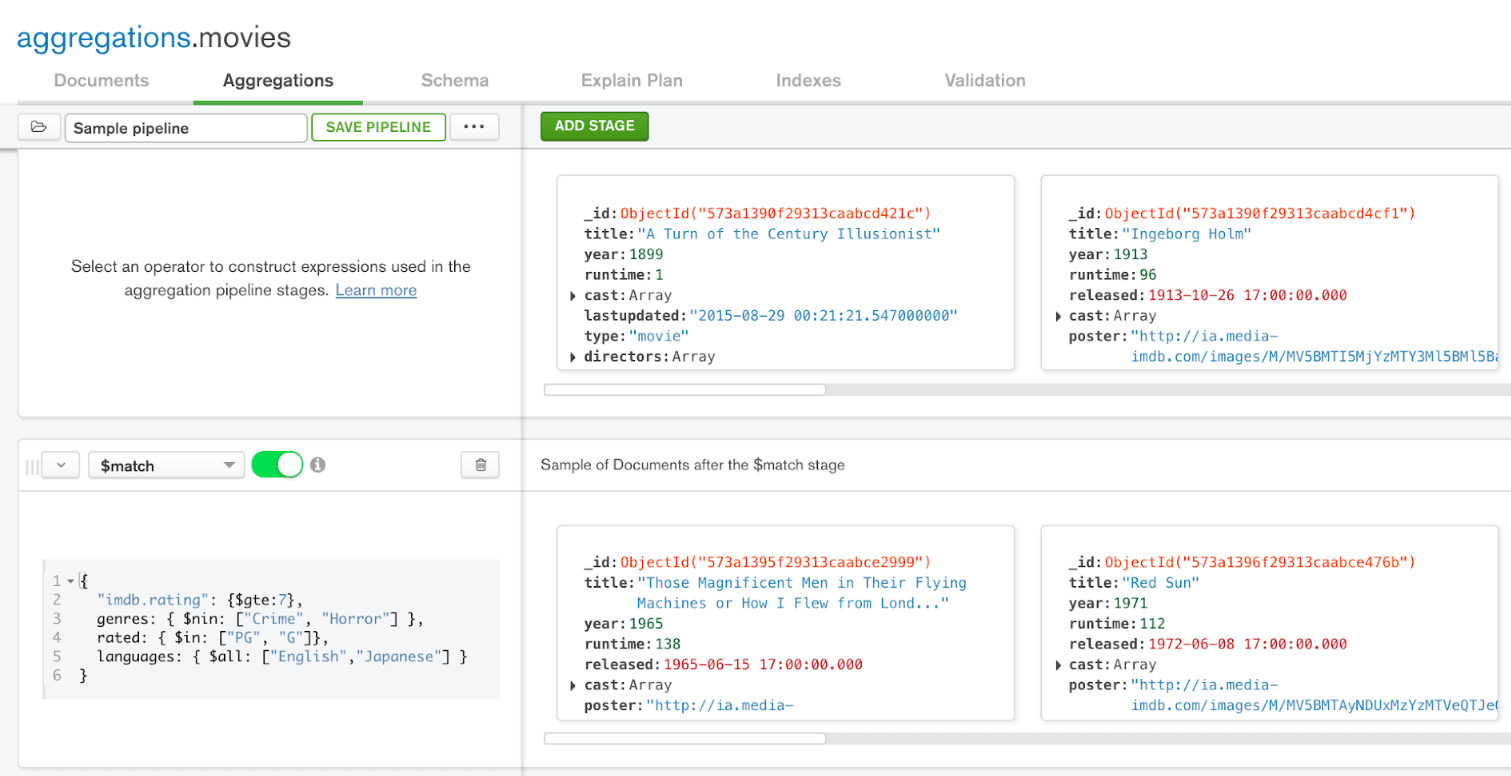
### **A simpler way to validate your data**

Write JSON Schema validation rules in an intelligent editor that auto-suggests field names, BSON data types, and validation keywords. With a live preview of documents that pass and fail the validation rule, it's easy to see whether the rule has the desired behavior.



### **Extensible via plugins**

The Compass Plugin Framework is exposed as an API, making it extensible by users. Looking for other functionality? Install a plugin or build your own.



### **Aggregations made easy**

Construct aggregation pipelines in an intuitive UI. Code skeletons and auto-complete make it easy to build a stage, while a preview of documents shows you if the stage is doing what you need. Add stages, remove them, or drag and drop to re-order in the pipeline. Once you’re done, export it to native code to use in your application.