MongoDB was **founded in 2007 by Dwight Merriman, Eliot Horowitz and Kevin Ryan** – the team behind DoubleClick. At the Internet advertising company DoubleClick (now owned by Google), the team developed and used many custom data stores to work around the shortcomings of existing databases. The business served 400,000 ads per second, but often struggled with both scalability and agility. Frustrated, the team was inspired to create a database that tackled the challenges it faced at DoubleClick. This was when MongoDB was born. MongoDB Atlas Fully managed, global cloud database on AWS, Azure, and GCP

The first version of the MongoDB database shipped in **August 2009**. The 1.0 release and those that followed shortly after were focused on validating a new and largely unproven approach to database design — built on a JSON-like document data model and layered onto an elastic and distributed systems foundation.

## What’s New in the Latest MongoDB Releases

* [2022 — MongoDB 6.0](https://www.mongodb.com/evolved#mdbsixzero)
* [2021-2022 — MongoDB 5.0 & Rapid Releases](https://www.mongodb.com/evolved#mdbfivezero)
* [2020 — MongoDB 4.4](https://www.mongodb.com/evolved#mdbfourfour)
* [2019 — MongoDB 4.2](https://www.mongodb.com/evolved#mdbfourtwo)
* [2018 — MongoDB 4.0](https://www.mongodb.com/evolved#mdbfourzero)
* [2017 — MongoDB 3.6](https://www.mongodb.com/evolved#mdbthreesix)
* [2016 — MongoDB 3.4](https://www.mongodb.com/evolved#mdbthreefour)
* [2015 (Late) — MongoDB 3.2](https://www.mongodb.com/evolved#mdbthreetwo)
* [2015 (Early) — MongoDB 3.0](https://www.mongodb.com/evolved#mdbthreezero)

### **2022 — MongoDB 6.0**

* [MongoDB 6.0](https://www.mongodb.com/blog/post/big-reasons-upgrade-mongodb-6-0) includes more features and optimizations for time series collections; improved support for event-driven architectures; full support for sharded joins and graph traversal; improvements to operational resilience and sharding; and the ability to [run expressive queries on fully randomized encrypted data](https://www.mongodb.com/products/queryable-encryption).
* General availability of [Atlas Serverless instances](https://www.mongodb.com/use-cases/serverless), [Atlas Data API](https://www.mongodb.com/docs/atlas/api/data-api/), [Atlas CLI](https://www.mongodb.com/docs/atlas/cli/stable/), and [Flexible Sync](https://www.mongodb.com/atlas/app-services/device-sync), which enables the cloud-to-edge synchronization of only the data that’s relevant to a given user or device.
* [Atlas Data Lake](https://www.mongodb.com/atlas/data-lake) (in preview), a fully managed storage service for analytical workloads; [Atlas Data Federation](https://www.mongodb.com/atlas/data-federation), which allows you to seamlessly query, transform, and aggregate data from one or more MongoDB Atlas databases, Atlas Data Lakes, or AWS S3 buckets; and new [Atlas SQL Interface](https://www.mongodb.com/atlas/sql) with support for popular SQL-based tools.
* [Cluster-to-cluster sync](https://www.mongodb.com/products/cluster-to-cluster-sync), which allows you to continuously synchronize data between MongoDB clusters in the same or hybrid environments, including Atlas, private cloud, on-premises, and at the edge.

[MongoDB release notes](https://www.mongodb.com/docs/upcoming/release-notes/?_ga=2.56519659.1300203159.1658758237-1881248372.1643815303)

### **2021-2022 — MongoDB 5.0 and Rapid Releases**

* [MongoDB 5.x](https://www.mongodb.com/collateral/mongodb-5-whats-new) with native [time series collections](https://www.mongodb.com/time-series) optimized for IoT and financial apps; live resharding so you can change your shard key on-demand with no database downtime; distributed cross-shard JOINs and graph traversals for sophisticated analytics against live data, faster initial sync via file copy, new aggregation operators, and more.
* The [MongoDB Stable API](https://docs.mongodb.com/v5.0/reference/versioned-api/) future-proofs your applications. You can upgrade to the latest MongoDB releases without the risk of backward-breaking changes.
* [Atlas Serverless instances](https://www.mongodb.com/cloud/atlas/serverless) (preview) automatically and dynamically scale to meet your workload and you pay only for the resources consumed.
* The [MongoDB Atlas Data API](https://docs.atlas.mongodb.com/api/data-api/) (preview) provides a fully managed, REST-like API for accessing your Atlas data without the need for database drivers.

[MongoDB release notes](https://docs.mongodb.com/upcoming/release-notes/)

### **2020 — MongoDB 4.4**

* [MongoDB 4.4](https://www.mongodb.com/collateral/mongodb-4.4-guide) offering richer aggregations with UNION; streaming replication reducing data synchronization latency across a distributed database cluster by up to 50% ; hedged and mirrored reads for consistent low latency in the face of infrastructure failures.
* [MongoDB Atlas Online Archive](https://www.mongodb.com/atlas/online-archive) to automatically tier aged data from your database to fully managed, queryable object storage, optimizing scalability, performance, and cost.
* [Realm & Sync](https://www.mongodb.com/realm), delivering best-in-class experiences at the edge of the network with an embedded mobile database and automated sync to MongoDB Atlas in the cloud, keeping data updated across users, devices, and your backend.
* [MongoDB Atlas multi-cloud clusters](https://www.mongodb.com/blog/post/introducing-multicloud-clusters-on-mongodb-atlas) providing the ability to distribute data in a single cluster across multiple public clouds simultaneously, or move workloads seamlessly between them.

[MongoDB release notes](https://docs.mongodb.com/upcoming/release-notes/)

### **2019 — MongoDB 4.2**

* [MongoDB 4.2](https://www.mongodb.com/collateral/mongodb-4.2-guide-to-what-is-new) brings distributed, cross-shard ACID transactions for data integrity at global scale; [client-side field-level encryption](https://www.mongodb.com/client-side-encryption), providing some of the strongest privacy controls anywhere; on-demand materialized views for blazing fast analytics.
* [MongoDB Atlas Search](https://www.mongodb.com/atlas/search), combining the power of Apache Lucene with the Atlas platform, making it easy to build fast, relevant, full-text search on top of your data in the cloud.
* [MongoDB Atlas Data Lake](https://www.mongodb.com/atlas/data-lake), enabling you to quickly and easily query data in any format on Amazon S3 using the MongoDB Query API.
* [MongoDB Operator for Kubernetes](https://www.mongodb.com/kubernetes) and [MongoDB Connector for Apache Kafka](https://www.mongodb.com/kafka-connector), simplifying MongoDB integration into your application estate.

[MongoDB release notes](https://docs.mongodb.com/manual/release-notes/)

### **2018 — MongoDB 4.0**

* MongoDB 4.0 offers [multi-document ACID transactions](https://www.mongodb.com/transactions), making it even easier to address a complete range of use cases with MongoDB and simplifying legacy database migrations.
* [MongoDB Atlas](https://www.mongodb.com/atlas/database) Global Clusters, creating fully managed, globally distributed database deployments for low-latency reads and writes, plus data placement controls for regulatory compliance.
* [MongoDB Atlas enterprise security controls](https://www.mongodb.com/cloud/trust) with LDAP integration; bring-your-own KMS for encrypting data at rest; and granular event audit logging.
* [MongoDB Charts](https://www.mongodb.com/products/charts) is a modern data visualization and analytics tool that allows you to easily create, share, and embed visualizations from Atlas and Atlas Data Lake.

[MongoDB release notes](https://docs.mongodb.com/manual/release-notes/)

### **2017 — MongoDB 3.6**

* [Fully managed MongoDB Atlas](https://www.mongodb.com/atlas/database) database service is now expanded from AWS to Azure and Google Cloud, providing unmatched data distribution across all of the leading cloud providers.
* [Change streams](https://docs.mongodb.com/manual/changeStreams/) to build always-on, real time, reactive applications and [retryable writes](https://docs.mongodb.com/manual/core/distributed-queries/" \l "retryable-writes" \t "_target) enabling developers to build more resilient apps with less client-side code.
* Further improved data integrity with [schema validation](https://docs.mongodb.com/manual/core/schema-validation/) to enforce a schema against your data.
* Implementation of a global logical clock to enforce consistent time across every operation in a distributed cluster, further improving data integrity and resilience, along with [causal consistency guarantees](https://docs.mongodb.com/upcoming/core/read-isolation-consistency-recency/#causal-consistency) for read-your-own-write consistency.

[MongoDB release notes](https://docs.mongodb.com/manual/release-notes/)

### **2016 — MongoDB 3.4**

* [Fully-managed MongoDB Atlas](https://www.mongodb.com/atlas/database) database service launched on AWS, providing built-in automation for resource and workload optimization and always-on security, backed by a 99.995% uptime SLA.
* Native graph processing with [$graphLookup](https://docs.mongodb.com/manual/reference/operator/aggregation/graphLookup/) to identify patterns in connected data; the [decimal data type](https://docs.mongodb.com/upcoming/release-notes/3.4/#decimal-type) for high-precision processing of financial and scientific data; and [read-only views](https://docs.mongodb.com/manual/core/views/) to filter and mask data.
* [Zoned sharding](https://docs.mongodb.com/upcoming/core/zone-sharding/) to localize data within specific regions and 10x faster data rebalancing across elastically scaled database clusters.
* [MongoDB Connector for Apache Spark](https://www.mongodb.com/products/spark-connector) providing seamless integration into data science and AI workflows.

[MongoDB release notes](https://docs.mongodb.com/manual/release-notes/)

### **2015 (Late) — MongoDB 3.2**

* The [Encrypted Storage Engine](https://docs.mongodb.com/manual/core/security-encryption-at-rest/#std-label-encrypted-storage-engine) provides native at-rest encryption without the performance or management overhead of separate file system encryption; the [In-Memory Storage Engine](https://docs.mongodb.com/manual/core/inmemory/) delivers high performance and predictable latency; and the [$lookup aggregation pipeline stage](https://docs.mongodb.com/manual/reference/operator/aggregation/lookup/) joins documents from different collections and databases.
* The launch of [MongoDB Compass](https://www.mongodb.com/products/compass) provides a GUI for MongoDB development and administration; the [MongoDB Connector for BI](https://www.mongodb.com/products/bi-connector) exposing MongoDB data for analysis and visualization via SQL.
* Higher database resilience with faster failure detection and recovery via the RAFT-based replication consensus protocol.

[MongoDB release notes](https://docs.mongodb.com/manual/release-notes/)

### **2015 (Early) — MongoDB 3.0**

* [MongoDB 3.0 with the WiredTiger Storage Engine](https://docs.mongodb.com/manual/core/wiredtiger/) offers document-level concurrency control and built-in compression for an order of magnitude more scalability.
* [MongoDB Ops Manager](https://www.mongodb.com/products/ops-manager) is the self-hosted management platform that enables you to deploy, monitor, back up, and scale MongoDB on your own infrastructure with 95% lower operational overhead.
* 50-member replica sets, providing global data distribution.

[MongoDB release notes](https://docs.mongodb.com/manual/release-notes/)

## The Impact of MongoDB

As time went on, MongoDB moved up the ranks to become the most popular type of database for document stores, and the fourth most popular database system overall. It is used by other highly successful companies like eBay, Abobe, LinkedIn, Foursquare, McAfee, Shutterfly, and others.

The most reliable way to recover deleted data is to use a Supported Backup Method with continuous backup (for example, agent-based backup like MongoDB Ops Manager) or an acceptable schedule for your recovery scenarios. Ops Manager also includes a Queryable backup feature to help with selective recovery of data

In comparison to the SQL server, MongoDB is faster and more scalable. While the SQL server supports JOIN and Global transactions, MongoDB does not. The MS SQL server does not accommodate large amounts of data, however MongoDB does.MongoDB is NoSQL



**MongoDB is a database based on a non-relational document model**. Thus, as a so-called NoSQL database (NoSQL = Not-only-SQL), it differs fundamentally from conventional relational databases such as Oracle, MySQL or the Microsoft SQL Server.

**MongoDB has been adopted as backend software** by a number of major websites and services including EA, Cisco, Shutterfly, Adobe, Ericsson, Craigslist, eBay, and Foursquare.

MongoDB uses the **MongoDB Query Language (MQL)**, designed for easy use by developers. The documentation compares MQL and SQL syntax for common database operations.

**MySQL** is a mature relational database system, offering a familiar database environment for experienced IT professionals.

**MongoDB** is a well-established, non-relational database system offering improved flexibility and horizontal scalability, but at the cost of some safety features of relational databases, such as referential integrity.

MySQL is faster at selecting a large number of records, while MongoDB is significantly faster at inserting or updating a large number of records.

#### **MySQL**

MySQL is a popular, free-to-use, and open-source relational database management system (RDBMS) developed by Oracle. As with other relational systems, MySQL stores data using tables and rows, enforces referential integrity, and uses structured query language (SQL) for data access. When users need to retrieve data from a MySQL database, they must construct an SQL query that joins multiple tables together to create the view on the data they require.

Database schemas and data models need to be defined ahead of time, and data must match this schema to be stored in the database. This rigid approach to storing data offers some degree of safety, but trades this for flexibility. If a new type or format of data needs to be stored in the database, schema migration must occur, which can become complex and expensive as the size of the database grows.

#### **MongoDB**

MongoDB is also free to use and open source; however, its design principles differ from traditional relational systems. Often styled as a [non-relational](https://www.mongodb.com/compare/https/www.mongodb.com/non-relational-database) (or NoSQL) system, MongoDB adopts a significantly different approach to storing data, representing information as a series of JSON-like documents (actually stored as binary JSON, or [BSON](https://www.mongodb.com/json-and-bson)), as opposed to the table and row format of relational systems.

MongoDB documents consist of a series of [key/value](https://www.mongodb.com/key-value-database) pairs of varying types, including arrays and nested documents; however, the primary difference is that the structure of the key/value pairs in a given collection can vary from document to document. This more flexible approach is possible because documents are self-describing.

 MongoDB also supports ACID properties of [transactions](https://www.mongodb.com/basics/transactions)

MongoDB Compass is a powerful GUI for querying, aggregating, and analyzing your MongoDB data in a visual environment.

Compass is free to use and source available, and can be run on macOS, Windows, and Linux.

 A database is a container for [collections](https://www.mongodb.com/docs/compass/current/collections/#std-label-collection-tab). Each database gets its own set of files on the host file system. A single MongoDB server typically has multiple databases. A collection is a grouping of MongoDB [documents](https://www.mongodb.com/docs/compass/current/documents/#std-label-compass-documents). Documents within a collection can have different fields. A collection is the equivalent of a table in a relational database system. A collection exists within a single [database](https://www.mongodb.com/docs/compass/current/databases/#std-label-database-tab)

## [Sharding](https://www.mongodb.com/docs/manual/reference/glossary/#std-term-sharding) is a method for distributing data across multiple machines. MongoDB uses sharding to support deployments with very large data sets and high throughput operations. Database systems with large data sets or high throughput applications can challenge the capacity of a single server.

## MongoDB Atlas provides an easy way to host and manage your data in the cloud.

NoSQL is preferable when you want flexibility like in startups where things may change in the future with respect to the opportunities they get as here it’s not bound to a particular structure. But for well-established companies where operations are fixed they can go for SQL DBs in general.

See the following difference between two most popular SQL and NoSQL Dbs-

MySQL MongoDB

1)Matured or stable Its new and updated frequently

2)It follows tabular structure It follows document structure like JSON format

3)It needs a proper schema Its flexible in nature

4)Managing complex relations among different tables is easy|Its not that great in complex managing relationship|

5)Its scales vertically Horizontaly scalable

Working with MongoDB

MongoDB is a document-based, general-purpose, distributed database with scalability and flexibility. And most of the features are free to use.

**Features of MongoDB**

**\*\*i. Rich JSON Documents-\*\***

\* The most natural and productive way to work with data.

\* Supports arrays and nested objects as values.

\* Allows for flexible and dynamic schemas.

\* The document model maps to the objects in your application code, making data easy to work with.

```JSON

{

 "name": "notebook",

 "qty": 50,

 "rating": [ { "score": 8 }, { "score": 9 } ],

 "size": { "height": 11, "width": 8.5, "unit": "in" },

 "status": "A",

 "tags": [ "college-ruled", "perforated"]

}

**Powerful query language-\*\***

\* Rich and expressive query language that allows you to filter and sort by any field, no matter how nested it may be within a document.

\* Support for aggregations and other modern use-cases such as geo-based search,  graph search, and text search.

\* Queries are themselves JSON, and thus easily composable. No more concatenating strings to dynamically generate SQL queries.

```

> db.collection.find( { qty: { $gt: 4 } } )

```

OUTPUT:

```JSON

{ "\_id": "apples", "qty": 5 }

{ "\_id": "bananas", "qty": 7 }

```

**All the power of a relational database, and more...**

\* Full ACID(Atomicity, Consistency, Isolation, Durability) transactions.

\* Support for joins in queries.

\* Two types of relationships instead of one: reference and embedded.

\* The fastest way to create visualizations of MongoDB data.

\* Built for the document model.

\* Visualize live data from any of your MongoDB instances. Available on MongoDB Atlas.

**Charts**

\* The fastest way to create visualizations of MongoDB data.

\* Built for the document model.

\* Visualize live data from any of your MongoDB instances. Available on MongoDB Atlas.

**BI Connector**

\* Allow any BI tool that can speak the MySQL protocol to work with your MongoDB data.

\* Leverage the BI tools your organization already uses.

\* Perform federated analytics, combining data from MongoDB and other databases.

**Compass**

 19.3.2 Installing MongoDB (Community Edition) [[Official Documents](https://docs.mongodb.com/manual/administration/install-community/)]

**For Windows**

STEP 1: Download the installer file from download center from [here](https://www.mongodb.com/try/download/community?tck=docs\_server)

1. Select the Version

2. Choose platform as windows

3. Select package as msi

4. Click on download

---

STEP 2: Run the installer that you have downloaded

1. Double click on the installer for mongoDB (which has .msi extension) from the location where you have downloaded the file

STEP 3: Follow along the installation wizard

1. Choose the setup type Complete or Custom (for advance users). The Complete setup option will choose the default location in your PC to install unlike Custom installation which allows you to choose the other location to install.

2. Select install MongoDB as a Service.

3.Select Run the service as Network Service user (Default)

4. Service Name. Specify the service name. Default name is MongoDB. If you already have a service with the specified name, you must choose another name.

5. Data Directory. Specify the data directory, which corresponds to the --dbpath. If the directory does not exist, the installer will create the directory and sets the directory access to the service user.

6. Log Directory. Specify the Log directory, which corresponds to the --logpath. If the directory does not exist, the installer will create the directory and sets the directory access to the service user.

STEP 4: Install [MongoDB compass](https://www.mongodb.com/products/compass) (Optional)

MongoDB compass is a UI tool for mongoDB. Once Compass is installed you'll see following screens -

**For Ubuntu**

Refer Official Docs - [here](https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/)

**For Mac**

Refer Official Docs - [here](https://docs.mongodb.com/manual/tutorial/install-mongodb-on-os-x/)

**Pymongo**

use the following command in your anaconda prompt to install pymongo

```

python -m pip install pymongo

```

Or you can directly run below cell to install pymongo in current environment-

!python -m pip install pymongo

**MongoDB Atlas**

\* Atlas is a mongoDB service on cloud. It uses AWS, Azure and GCP cloud services to cater developers all around the globe for managing mongoDB databases.

\* Its a globally available cloud database service for all kinds of modern applications.

\* Visit [ATLAS HOMEPAGE](https://www.mongodb.com/cloud/atlas) for more details.

Data distribution system is master slave architecture where data is on Multiple system at a time. we can scaleup easily, add data without schema ,query search are faster uses index search.jupitor notebook is web based python editor,never execute the code,never interpret your code.it uses key value pair.it uses collection bunch of heterogeneous dataset .power bi can import data through adirect connection to the mongodb bi connector via odbc once a data connection has been defined simply select the data you want to work with and import it. Mongosyphon is a specially desined etl tool to transform data into mongo document structure.it can read and extract data from rdbms table convert into json document,xml output or write directly on mongodb.

<https://github.com/MKLab-ITI>

<https://imgs.developpaper.com/imgs/1601821-20220416155546611-966519806.png>

<https://github.com/sudh9931/Big-Data-installation/blob/main/big%20data%20installation.txt>

<https://www.cloudera.com/downloads/hortonworks-sandbox/hdp.html>

[https://static.googleusercontent.com/media/research.google.com/en//archive/gfs-sosp2003.pdf](https://static.googleusercontent.com/media/research.google.com/en/archive/gfs-sosp2003.pdf)

<https://github.com/ritesh?tab=repositories>

<https://www.youtube.com/watch?v=eK8GjfwGXWQ>

<https://www.cloudera.com/downloads/hortonworks-sandbox/hdp.html>

Open alerts = var maxdate =maxx(dm\_tm,dm-tm[tm\_skey])

Return

Calculate(distinctcount(fact\_aed\_alert[alert\_id]), fact\_aed\_alert[period\_oc]<=maxdate ,fact\_aed\_alert[alert\_status\_description] IN {“open”,”monitor”})\*/

Var maxdate=calculate(max(fact\_aed\_alert[period\_oc]),ALL(dm\_cust))return

calculate(distinctcount(fact\_aed\_alert[alert\_id]),fact\_aed\_alert[period\_oc]