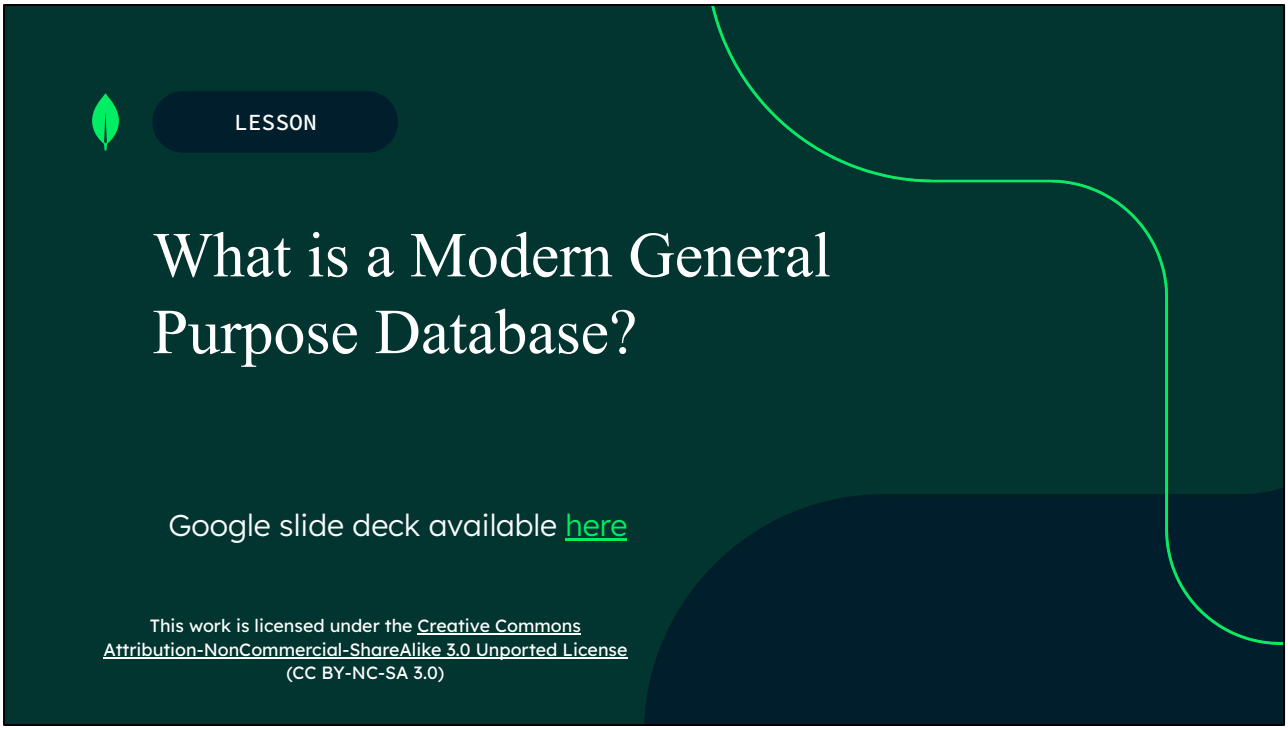

LESSON
What is a Modern General Purpose Database?
Google slide deck available here
This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License (CC BY-NC-SA 3.0)

What is a general purpose database?

Designed to meet the needs of as many applications as possible

Not designed to manage any specific type of data or workload

Influenced the design of many relational databases such as Oracle, MySQL, and Postgres

General purpose databases as a concept have been around since the 1960s. They are designed to meet the needs of a wide variety of applications. They are not just designed for a specific type of workload (e.g. analytics) or type of data (e.g. graphs).

The requirement to support a wide variety of applications has influenced the design of most relational databases including Oracle, MySQL, and Postgres.

MongoDB and the majority of non-relational databases were influenced by the requirement to support a variety of workloads, additionally they also were built to better scale horizontally as this was a known problem for relational databases during the time that non-relational databases were first being developed.

Let’s look at one of the larger factors in any database, the data size and what growing data has meant in terms of the database.

Growing Data Sizes

General purpose databases were designed to run on a single server and host the associated data on that server

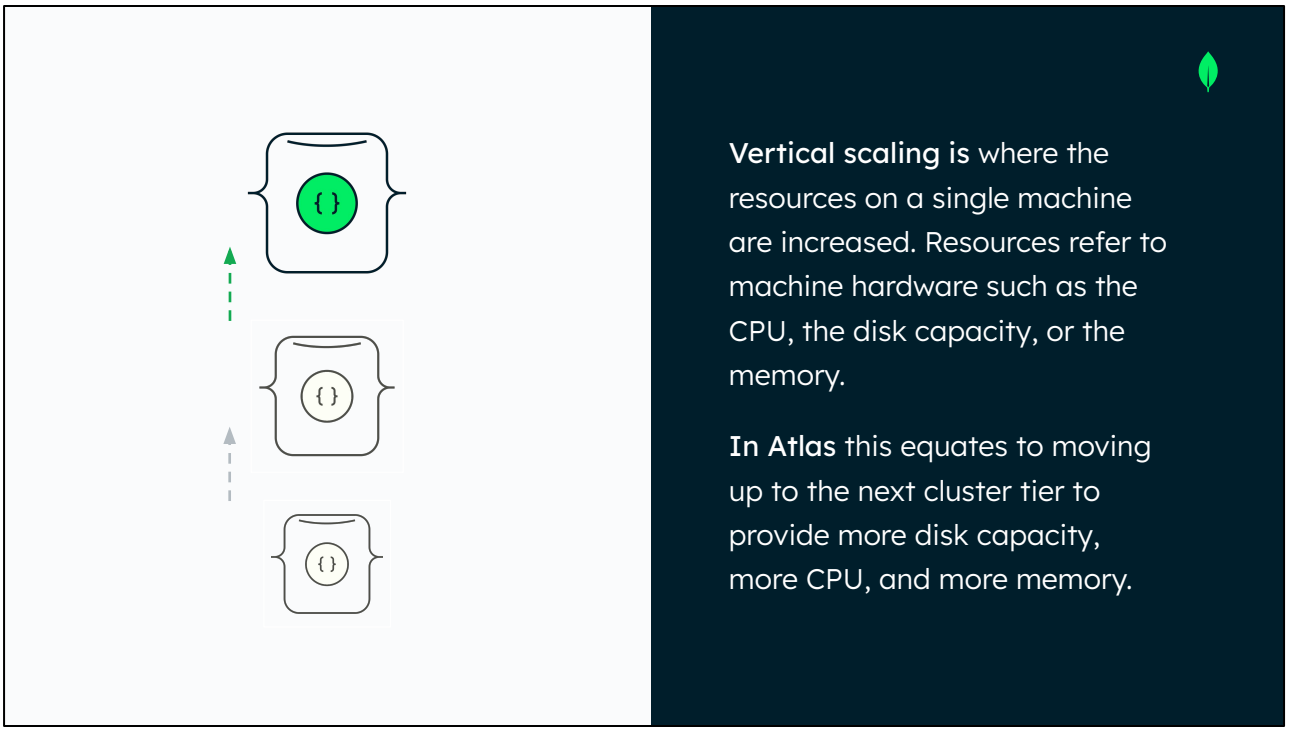
Exponential growth in data occurred with availability of both the Internet and of smart devices

Scaling and managing that scaling became a genuine problem for databases as data volumes increased

A major change occurred in the last three decades in computing and that is the size of data being stored. Over a relatively short period of time, the field of computing saw exponential growth in the size of data needing storage.

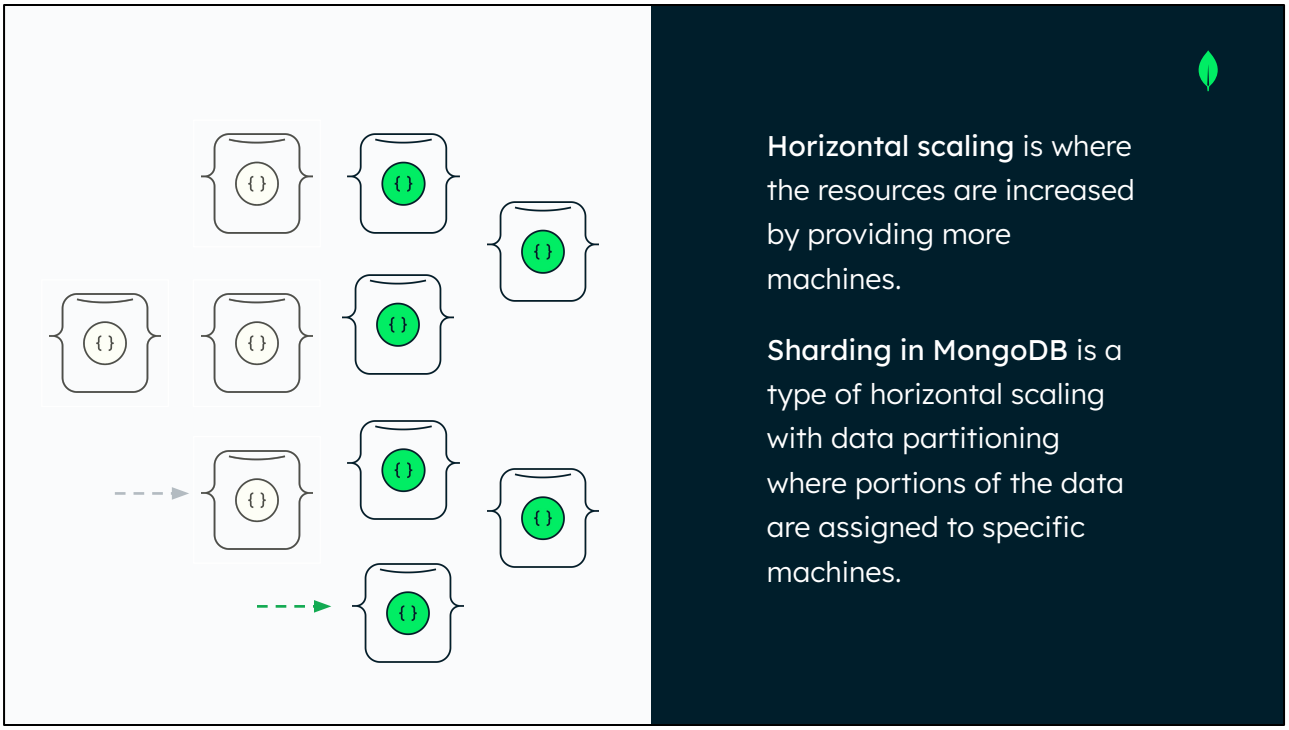
General purpose databases were originally designed to run and serve data from a single server, this relates back to the design origins in the 1960's and more recently as there were performance benefits to locate data on a single machine.

The growth of the Internet and indeed the massive numbers of smart devices all of which have one or more applications generating or consuming data or indeed both have added to this data growth.



Vertical scaling is where you incrementally increase the resources on a given machine such as CPU, disk, or memory to help scale. In Atlas this equates with moving to the next cluster tier.

In general, vertical scaling will have a finite limit as there will be a point where you can no longer keep adding resources on a single machine to solve the scaling problem. This can be addressed by horizontal scaling (which can also be done in conjunction with vertical scaling).



Horizontal scaling is where the data in the database is spread across many machines. In the case of MongoDB, this is achieved by sharding where the data is partitioned and each portion of the data is assigned to a specific shard, typically each shard will be backed in production by a replica set.

Software as a Service (SaaS)

SaaS is centrally hosted and paid in a subscription fashion.

Pioneered by Salesforce, this system had a major impact on software where previously enterprise software companies were the main providers of software.

Database as a Service (DBaaS)

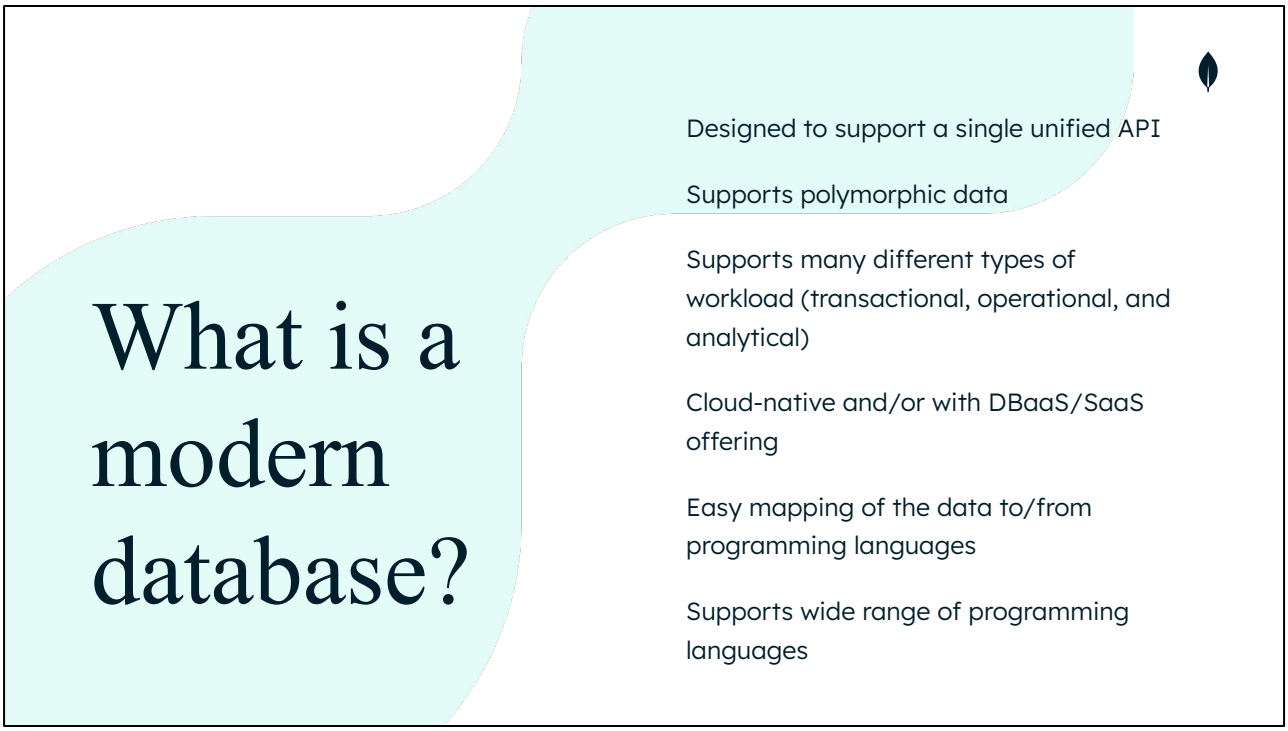
An example of a DBaaS is MongoDB's Atlas.

SaaS is where software is hosted centrally and you as a consumer/user of the software pays a subscription to use it. You don't have to run it on your machines as the SaaS provider deals with the hosting, maintenance, and general running of the software.

Salesforce is the noted pioneer in the space of SaaS. This delivery model has significantly changed the software industry where once enterprise software companies would have major versions released on an annual or longer period, SaaS companies now deliver new releases in days or weeks rather than in years.

In the database space, SaaS software that is focused on the provision of database and related hosting is often called DBaaS or Database as a Service.

This delivery velocity has changed the mindset in many software companies, including MongoDB which started as a enterprise software company and has moved increasingly into the SaaS space with it's database offering, Atlas. We'll discuss more on Atlas later.



There are several aspects to classifying a database as a modern database.

Firstly, it needs to support a single unified Application Programming Interface or API for short.

Next, it should support polymorphic data or data which can have many different shapes but which is still stored together.

A modern database should be able to support a variety of workloads whether analytical, operational, or transactional.

The changing infrastructure landscape and particularly that of the public cloud, whether cloud-native or DBaaS/SaaS are supported by a modern database.

It should also support the easy (seamless) mapping of data to and from programming languages.

Finally, a modern database supports a wide range of programming languages that can interface with it.

What is a modern general purpose database?

Combines all the previous features from both a modern database and from a general purpose database

In addition:

Can be containerized

Can be provisioned and run in multi cloud provider environments

Can support the geolocation of data

If we combine the concepts of a modern database and a general purpose database, we can find a few additional aspects that complete the definition for a modern general purpose database.

Building on the Cloud-native aspect, a modern general purpose database should be runnable in a container (as previously happened with virtual images in the era of virtualization for servers). It should also run in multiple cloud provider environments (it should not be tied to a specific provider or locked-in to a specific provider). A container or more specifically, a container image is ready-to-run software package, containing everything needed to run an application: the code and any runtime it requires, application and system libraries, and default values for any essential settings.

A final aspect is that it should be possible to locate the data or portions of the data within the database to specific machines (which can be located in specific locales). This ability is increasingly important with data privacy and legislation around data handling. In the US, the California Privacy Act (CCPA) is a good example whilst in Europe (specifically in the EU) the General Data Protection Regulation (GDPR) is a similar law. These laws require more thought when designing systems where users and data may be created/consumed/service a global audience.

Quiz

Quiz

Fill in the blanks to complete the sentences.

A modern general purpose database should provide a single unified _____.

A modern general purpose database should support _____ data or many types of data.

A modern general purpose database should be _____-native or provide a __aaS or _aaS offering.

In this quiz, you should fill in the blank for the three questions:

1. A modern general purpose database should provide a single unified …….?
2. A modern general purpose database should support _____ data or many types of data?
3. A modern general purpose database should be _____-native or provide a __aaS or _aaS offering.

Quiz

Fill in the blanks to complete the sentences.

A modern general purpose database should provide a single unified _____.

A modern general purpose database should support _____ data or many types of data.

A modern general purpose database should be _____-native or provide a __aaS or _aaS offering.

Fill in the blank - A modern general purpose database should provide a single unified _____.

Quiz

Fill in the blanks to complete the sentences.

A modern general purpose database should provide a single unified **application programming interface (API)**.

A modern general purpose database should support _____ data or many types of data.

A modern general purpose database should be _____-native or provide a __aaS or _aaS offering.

Modern general purpose databases should provide a unified API or application programming interface to allow the database to be effectively and programmatically used by software engineers and developers.

Quiz

Fill in the blanks to complete the sentences.

A modern general purpose database should provide a single unified application programming interface (API).

A modern general purpose database should support _____ data or many types of data.

A modern general purpose database should be _____-native or provide a __aaS or _aaS offering.

Fill in the blank - A modern general purpose database should support _____ data.

Quiz

Fill in the blanks to complete the sentences.

A modern general purpose database should provide a single unified application programming interface (API).

A modern general purpose database should support **polymorphic** data or many types of data.

A modern general purpose database should be _____-native or provide a __aaS or _aaS offering.

A modern general purpose database should support polymorphic data where many schemas can co-exist. A single schema can limit developer velocity and add hurdles when iteratively developing as schema migration is a requirement of normal development. The static nature of a single schema and the processes required in relational databases make schema migration a lengthy process that can add substantially to development time. Polymorphic data avoids this additional overhead as a key underpinning of this category of database is to enable the developer and their development velocity.

Quiz

Fill in the blank words to complete the sentences.

A modern general purpose database should provide a single unified application programming interface (API).

A modern general purpose database should support polymorphic data or many types of data.

A modern general purpose database should be _____-native or provide a __aaS or _aaS offering.

Fill in the blank - A modern general purpose database should be _____-native or provide a __aaS or _aaS offering.

Quiz

Fill in the blank words to complete the sentences.

A modern general purpose database should provide a single unified application programming interface (API).

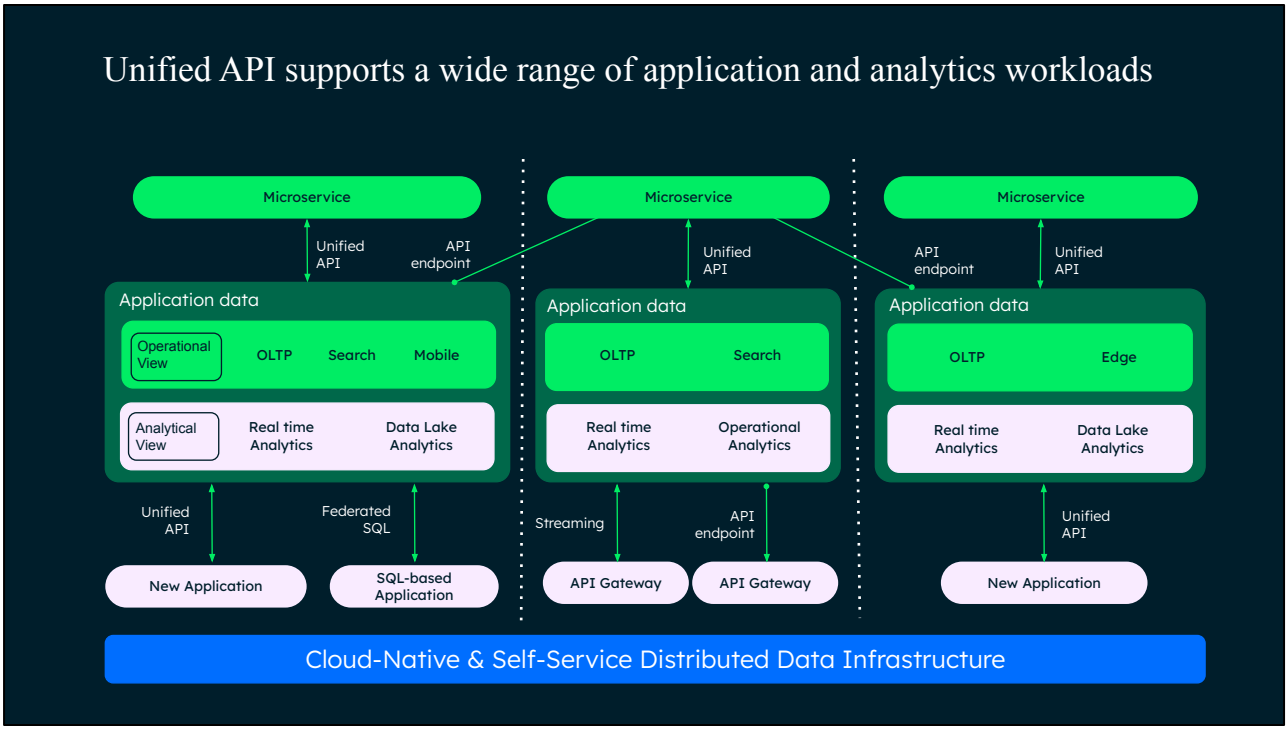
A modern general purpose database should support polymorphic data .or many types of data.

A modern general purpose database should be **Cloud-native** or provide a **DBaaS** or **SaaS** offering.

A modern general purpose database should be Cloud-native or provide a Database-as-a-Service or a Software-as-a-Service to effective support modern development approaches which use public cloud infrastructure. In the past, hosting your database and your own infrastructure was the only approach, now with public cloud infrastructure and the related tooling which enabled developers to more easily build and deploy their applications. This is a key differentiator of ‘modern’ general purpose databases.

MongoDB as an Example of a Modern General Purpose Database

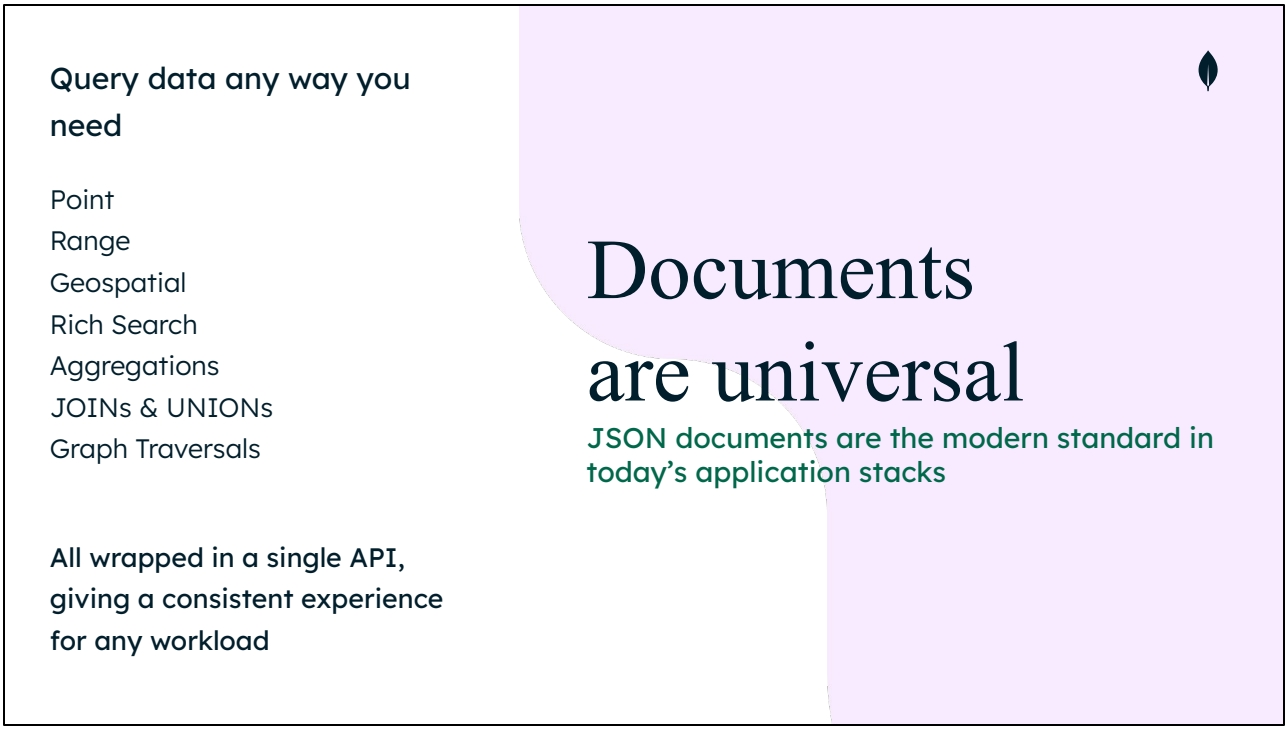
In the next slides, we will focus on MongoDB and highlight how it fulfills the various aspects of a modern general purpose database.



An application programming interface (API) is a feature of modern databases. MongoDB has a single standardised API, it allows for the database, Atlas Search and Atlas Data Lake to be programmatically accessed with the same calls and syntax.

The same application programming interface can be used to access/configure/or use Atlas clusters, Atlas's Data Lake or Atlas's Search functionality.

All of the current products and services hosted by MongoDB are broadly covered in the category of MongoDB Cloud. The purpose of this diagram is to highlight where new features or products have been added to the MongoDB data platform. A key realisation or takeaway is that for many SaaS/DBaaS companies that providing hosting of the database is no longer sufficient in terms of competitive features when developers are evaluating options in the DBaaS space.



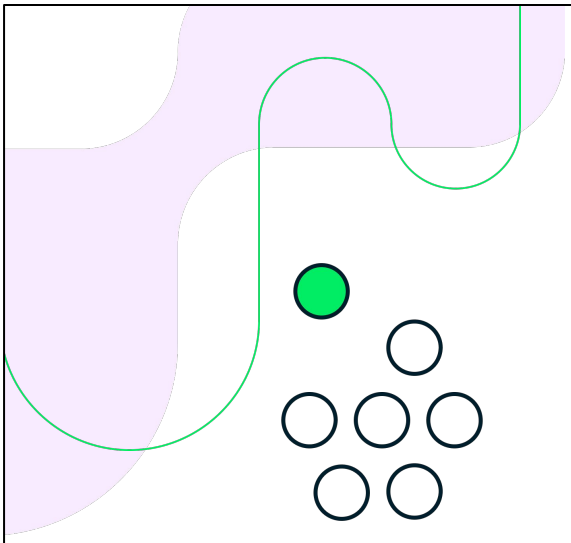
Documents are for modeling and querying in the manner you want to perform them.

There are a variety of data types and structures that can be used to hold data. Documents are a natural way to think about and model things, that's why MongoDB was created and why there is such a strong resonance with developers about how to model with it.

This allows point, range, or other geospatial data to be easily queried using either aggregations or rich search criteria. We'll cover this in more depth in the MQL and in the aggregation framework lessons.

In MongoDB, there are a range of data types in documents that can be supported. These are all wrapped into a single API and ensure that it is a consistent experience when designing applications for any kind of workload.

JSON documents are ubiquitous in modern development / application stacks and as a modern standard are universal in terms of their familiarity to developers.



Workload Isolation

A great advantage of MongoDB's distributed systems architecture is the ability to bring analytical workloads right alongside transactional workloads in the same scale-out cluster, providing fresher insights over vast data sets without all of the complexity and cost of data movement to independent analytics systems

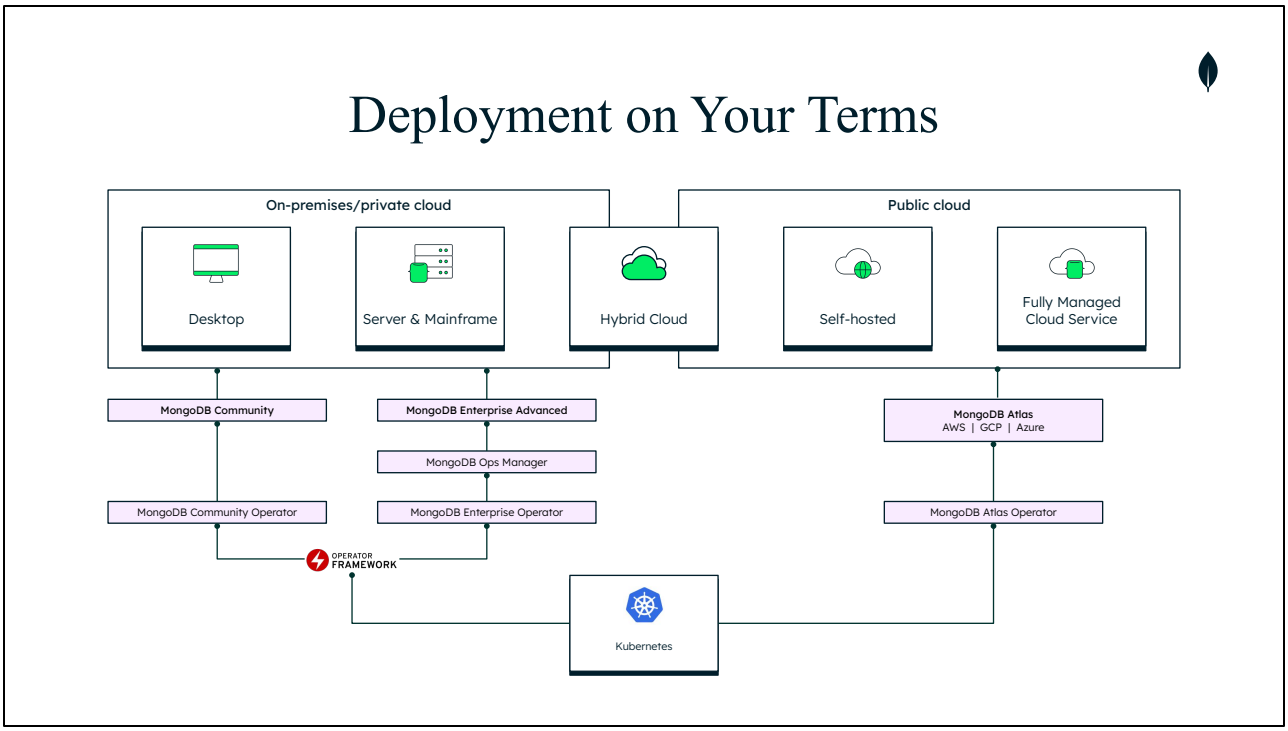
Being able to run operational workloads along with analytical workloads on the same platform is a big advantage as it means you do not need to provision a separate database or set of resources to service your analytical queries. This is done by designating a specific set of nodes to be read-only and which will service the analytical queries.

In the case of MongoDB, you can easily use tags to label nodes and use these tags with your applications to direct traffic to analytical nodes so that both your normal production traffic can be serviced as well as more ad-hoc analytical queries. The isolation ensures that query performance on your primary is not impacted because you are not changing the working set of data as the queries for any analytics are made on the designated analytics nodes. MongoDB Atlas supports this.

MongoDB offers a wide range of drivers for various programming languages as shown on the slide. The MongoDB Drivers are all built to meet various driver specifications, these specifications are available on Github at https://github.com/mongodb/specifications and you can find more details at https://docs.mongodb.com/drivers/specs on the various drivers directly supported by MongoDB. There are more programming languages with community supported drivers.

As all of the drivers implement the same specifications they all provide:

- Common CRUD capabilities but idiomatic to each language
- Uniform High Availability & Failover capabilities



In order to facilitate being a modern database that supports containerization and provisioning to public cloud providers MongoDB has a number of options. These MongoDB's Kubernetes integrations allow customers to run and scale clusters with ease regardless of their chosen infrastructure topology. This is the approach MongoDB has taken to support running cloud-native databases.

These operators allow you to seamlessly integrate MongoDB Atlas into your current Kubernetes deployment pipeline for a consistent experience across different deployment environments. Leave your workflow uninterrupted using the MongoDB Atlas Operator to simplify deployment, management and scaling of your Atlas clusters in Kubernetes.

With the Atlas Operator, developers can manage Atlas directly from the Kubernetes API to allow for simple and quick cluster and database user configuration so they can easily deploy and manage standardized clusters in any type of environment. The Atlas Operator supports most resources available in the MongoDB Atlas API, including projects, clusters, database users, IP access lists, network peerings, and more. For a complete list, see the Atlas Operator documentation.

A modern general purpose database offers:

A unified API

Support for a variety of workloads

Support for a variety of programming languages

Support for natural modelling of the concepts/objects

An easy mapping to constructs in the programming language

Capability with a variety of data types

The ability to be containerized

Capability to be provisioned and run in multi cloud provider environments

Ability to geo-locate data

TTL Indexes
Single Field indexes, when expired delete the document

Unique Indexes
Ensures value is not duplicated

Partial Indexes
Expression based indexes, allowing indexes on subsets of data

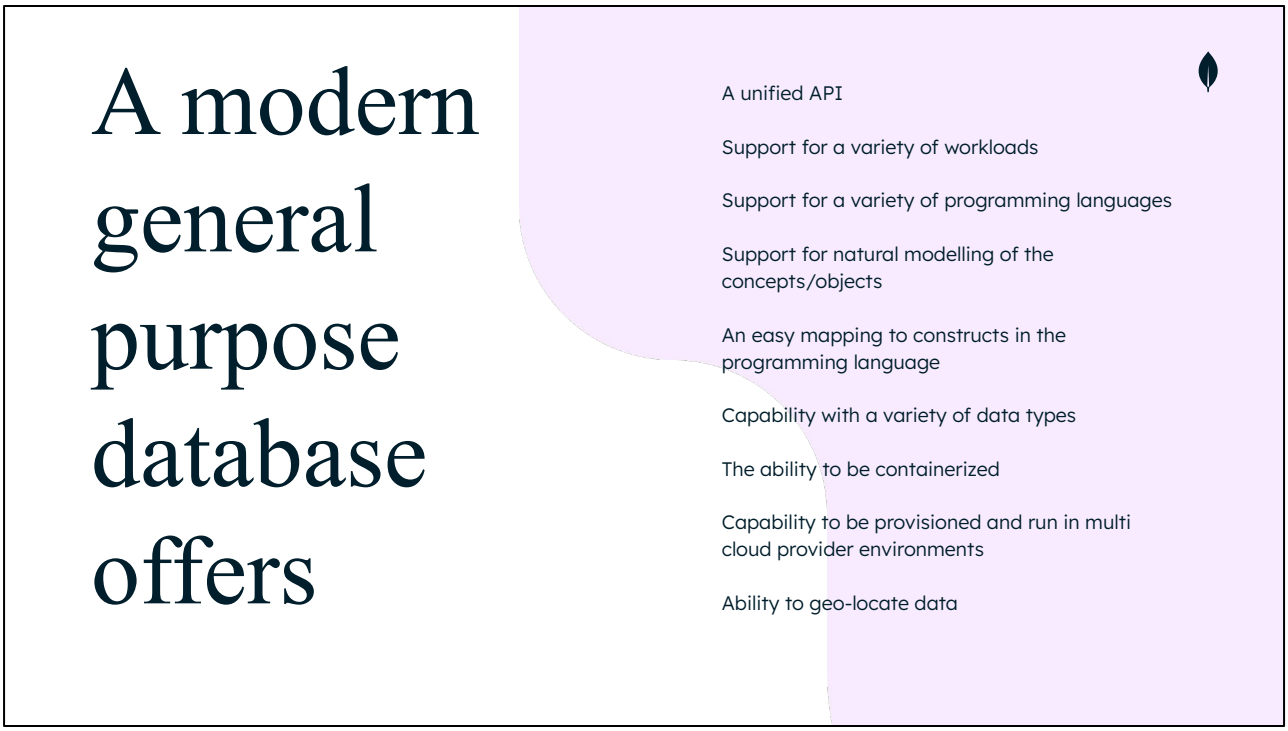
Case Insensitive Indexes
Supports text search using case insensitive search

Sparse Indexes
Only index documents which have the given field

A unified API, support for both a range of workloads and a range of programming languages with easy mapping are some of the core elements to a modern general purpose database.

Secondary indexes and a range of specialised indexes are also key to being a modern general purpose database as it allows flexibility in terms of performant queries whether this be with TTL, Unique, Partial, Sparse or indeed case insensitive indexes.

These are all elements to what constitutes a successful modern general purpose database.



Recapping our earlier slide, let's revisit the full list of features/functionality a modern general purpose database should offer:

- A consistent application programming interface (API) across versions and the platform/products
- The ability to handle a wide variety of workloads from analytical to transactional
- The ability to be programmed by a wide variety of programming languages
- A natural mapping of the data in the database to the programming concepts
- Support for a wide number of data types to provide granular storage of data in the most appropriate data type
- The ability for the database to be run in a container
- The ability for the database to be easily provisioned and run across the major public cloud environments
- The ability for the database to ensure that specific data can be stored in specific geographic locations/hardware

Quiz

Quiz

Which of the following are key features a modern general purpose database should support? *Select all that apply.* More than one answer choice can be correct.

- A. Unified API
- B. Various workloads
- C. Containerization
- D. Data geo-location for where it is stored
- E. Structured data

Quiz

Which of the following are key features a modern general purpose database should support? *Select all that apply.* More than one answer choice can be correct.

- ✓ A. Unified API
- ✓ B. Various workloads
- ✓ C. Containerization
- ✓ D. Data geo-location for where it is stored
- ✗ E. Structured data

CORRECT: Unified API - All modern general purpose databases should provide a consistent API that unifies programmatic access to the various features and functionality of the database.
CORRECT: Support for various workloads - Supporting analytical, transactional, and other workloads are a key feature to support a wide variety of applications and use cases.
CORRECT: Containerization - The ability for the database software to be containerized is growing increasingly more important, it follows from the earlier trend of virtualization. The requirement for a database to be easily run on multi tenant hardware to allow effective utilise resources is the driving rationale behind this.
CORRECT: Data geo-location for where it is stored. - This is correct. This supports many legal/data protection regulations but more importantly it can ensure that data is closest to the users who need/use it.
INCORRECT: Structured data. - This is incorrect. Structured data is not a required key feature of such a database as flexible data is the key to support general purpose workloads and data.

Quiz

Which of the following are key features a modern general purpose database should support? *Select all that apply.* More than one answer choice can be correct.

- ✓ A. Unified API
- ✓ B. Various workloads
- ✓ C. Containerization
- ✓ D. Data geo-location for where it is stored
- ✗ E. Structured data

This is correct. A modern general purpose databases should provide a consistent API that unifies programmatic access to the various features and functionality of the database.

CORRECT: Unified API - All modern general purpose databases should provide a consistent API that unifies programmatic access to the various features and functionality of the database.

Quiz

Which of the following are key features a modern general purpose database should support? *Select all that apply.* More than one answer choice can be correct.

- ✓ A. Unified API
- ✓ B. Various workloads
- ✓ C. Containerization
- ✓ D. Data geo-location for where it is stored
- ✗ E. Structured data

This is correct. Supporting analytical, transactional, and other workloads are a key feature to support a wide variety of applications and use cases.

CORRECT: Various workloads - Supporting analytical, transactional, and other workloads are a key feature to support a wide variety of applications and use cases.

Quiz

Which of the following are key features a modern general purpose database should support? *Select all that apply.* More than one answer choice can be correct.

- ✓ A. Unified API
- ✓ B. Various workloads
- ✓ C. Containerization
- ✓ D. Data geo-location for where it is stored
- ✗ E. Structured data

This is correct. The ability for the database software to be containerized is important as it allows a database to be easily run on multi tenant hardware for effective utilise resources.

CORRECT: Containerization - The ability for the database software to be containerized is growing increasingly more important, it follows from the earlier trend of virtualization. The requirement for a database to be easily run on multi tenant hardware to allow effective utilise resources is the driving rationale behind this.

Quiz

Which of the following are key features a modern general purpose database should support? *Select all that apply.* More than one answer choice can be correct.

- ✓ A. Unified API
- ✓ B. Various workloads
- ✓ C. Containerization
- ✓ D. Data geo-location for where it is stored
- ✗ E. Structured data

This is correct. This supports many legal/data protection regulations but more importantly it can ensure that data is closest to the users who need/use it.

CORRECT: Data geo-location for where it is stored. - This is correct. This supports many legal/data protection regulations but more importantly it can ensure that data is closest to the users who need/use it.

Quiz

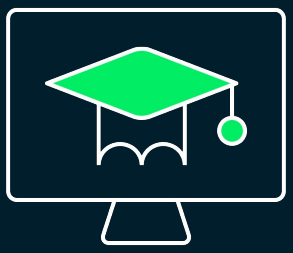
Which of the following are key features a modern general purpose database should support? *Select all that apply.* More than one answer choice can be correct.

- ✓ A. Unified API
- ✓ B. Various workloads
- ✓ C. Containerization
- ✓ D. Data geo-location for where it is stored
- ✗ E. Structured data

This is incorrect. In examples, documents are often represented in JSON, but under the hood, MongoDB uses BSON which offers more data types and better performance.

This is incorrect. In examples, documents are often represented in JSON, but under the hood, MongoDB uses BSON which offers more data types and better performance.

Continue Learning!



MongoDB University has free self-paced courses and labs ranging from beginner to advanced levels.

Github Student Developer Pack

Sign up for the MongoDB Student Pack to receive $50 in Atlas credits and free certification!

This concludes the material for this lesson. However, there are many more ways to learn about MongoDB and non-relational databases, and they are all free! Check out MongoDB's University page to find free courses that go into more depth about everything MongoDB and non-relational. For students and educators alike, MongoDB for Academia is here to offer support in many forms. Check out our educator resources and join the Educator Community. Students can receive $50 in Atlas credits and free certification through the Github Student Developer Pack.