

# Exploration — Introduction to Database Management Systems

## Introduction



Practically all web applications need to store data persistently. Static files, e.g., HTML pages, JavaScript code, CSS stylesheets, image files, etc., can be stored on the file system. However, beyond the simplest web applications, all web applications also have a need for a database management system. In this exploration, we first briefly discuss different types of database management systems and then introduce MongoDB, the database management system we will use in this course.

## Types of Database Management Systems

A **database** is an organized collection of data while a **database management system** (DBMS) is the software used to manage databases. Many times, when the context is clear, the term database is used for DBMS as well. A DBMS supports efficient CRUD (create, read, update, delete) operations on large volumes of data. Another important feature that DBMSs provide is the concurrent execution of CRUD operations for a large number of users. We can categorize DBMSs along many different dimensions. Two dimensions are most relevant for us at this point and we discuss them below:

### 1. What abstractions does the DBMS provide for modeling data?

Relational DBMSs	Document DBMSs
Table	Collection
Row	Document
Column	Property

### Modeling data in Relational DBMSs and in Document DBMSs

**Relational DBMS**, such as Oracle, SQL Server, PostgreSQL, MySQL, etc., are the most common types of DBMS in current use. Commercial relational DBMSs appeared around early 1980's and have dominated the DBMS market since the late 1980's.

- In a relational DBMS, data is organized as **tables**.
- The **columns** of a table define the data type of the data that can be stored in that table.
- The **rows** of a table store the actual values corresponding to that column.
- Relational DBMS technology is very mature with decades of research and development.
- Over the years, standards have been defined for relational technology. Due to this, if you have written applications using one relational DBMS, there is an easy learning curve for writing applications for another relational DBMS.

**Document-oriented DBMS**, such as MongoDB, Amazon's DocumentDB, Couchbase, Google's Firestore, are comparatively new. These DBMS have gained some popularity starting around 2005.

- In a document-oriented DBMS, also simply called a **document DBMS**, data is stored as a **document** in a format such as JSON or XML.
- Documents that are somewhat similar are added to the same **collection**.
- A document corresponds to a row in a relational DBMS, while a collection corresponds to a table.
- A **property** of a document corresponds to a column in a table.
- Most document DBMS do not require that the data type of properties should match across the different documents stored in the same collection.
- There are not many widely used standards for document DBMS. Switching from one document database to another can be a steep curve.

There are a few other types of DBMS being used in industry that provide different abstractions for modeling data. Prominent among these are Key-value DBMS and Graph DBMS. But we will not discuss these further in this course.

## 2. What language or API does the DBMS provide for CRUD operations on the data?

---

Relational DBMS support a standard query language called **SQL** (Structured Query Language). SQL provides support for all CRUD operations on tables, as well as support for creating, updating and dropping tables. There are only minor variations in the SQL syntax across different DBMS, which again makes it easy to transfer your knowledge of developing applications for one relational DBMS to developing apps for another relational DBMS.

Document DBMS started out as explicitly rejecting SQL and were associated with a movement termed **NoSQL** DBMS. CRUD operations were, and continue to be, supported via functions provided by the document DBMS. However, over time many document DBMS recognized the usefulness of query languages and many now support some form of query language. However, the lack of standardization means that the query languages of different document DBMS can vary a lot.

## Choosing a DBMS

---

When developing applications, choosing a type of DBMS and then a specific DBMS within that type of DBMSs is an extremely important architectural choice. This choice requires taking into consideration many factors, including the scale and performance needs of the application, maintenance and availability requirements, geographical distribution of the users, as well as other considerations. This is a topic far beyond the scope of this course.

In this course we will use MongoDB which is a document DBMS which doesn't support SQL as a query language. The format of documents in MongoDB closely resembles JSON. More specifically, MongoDB internally stores documents in a format called BSON, which is a binary form of JSON. But the MongoDB API exposes the documents as JSON and the JSON format makes it very easy to use MongoDB in Node programs.

## Setting Up MongoDB

We will use MongoDB 5 Community Edition in this course. One way to use MongoDB is to install and run it on our local machine. Directions for doing this for various operating systems, including **Windows** (<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/#install-mongodb-community-edition>) and **Mac** (<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/#install-mongodb-community-edition>) are available at the **Install MongoDB directions page** (<https://docs.mongodb.com/manual/installation/>). Another alternative is to use a hosting service for running MongoDB. MongoDB Inc., the company behind the MongoDB DBMS, provides a service called MongoDB Atlas for running MongoDB. There are various options for this, including paid and a free service. **We recommend using MongoDB Atlas service**, rather than installing MongoDB on your machine (which can be **error-prone, especially on Mac OS**). (NOTE: A small percentage of students cannot connect to Atlas due to their Internet Service Provider blocking access. You may have to update the settings of your Router. In this case, consider installing MongoDB locally.) And, another note: you will need your IP address, which you can find using <https://www.whatismyip.com/> (<https://www.whatismyip.com/>).

<a href="https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/#install-mongodb-community-edition">https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/#install-mongodb-community-edition</a> <b>Atlas Cloud Cluster</b> <a href="https://canvas.oregonstate.edu/courses/1879154/files/94412058?wrap=1">https://canvas.oregonstate.edu/courses/1879154/files/94412058?wrap=1</a> for Mac, Windows, Linux	<b>Windows Installation</b> <a href="https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/#install-mongodb-community-edition">https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/#install-mongodb-community-edition</a> on the local Har
--	---

## MongoDB on Atlas Cloud Cluster

MongoDB Inc., the company behind the MongoDB DBMS, provides hosting services for running MongoDB servers. In addition to paid services, they also have a free offering that you can use to run a MongoDB server. In the following video, we show how to create a MongoDB server on Atlas Cloud. Follow along with the video to create a cluster for your use in the class, or follow **Setup**

## MongoDB using Atlas Cloud Cluster

(<https://canvas.oregonstate.edu/courses/1879154/files/94412058?wrap=1>).(PDF).

IMPORTANT: Close the connection with your localhost:27017 and restart VS Code to clear any confusion between your app and the Atlas Cluster.



## Connecting Our Node App to a MongoDB Server

Let's now test whether we have everything setup correctly for Node apps running on our machine to connect to a MongoDB server. Numerous npm packages are available for use in Node.js apps to connect to a MongoDB server. We will use the npm package Mongoose in this course. Here is a [MongoDB Atlas Setup Test.zip](https://canvas.oregonstate.edu/courses/1879154/files/93831999?wrap=1) (<https://canvas.oregonstate.edu/courses/1879154/files/93831999?wrap=1>) [↓](https://canvas.oregonstate.edu/courses/1879154/files/93831999/download?download_frd=1) ([https://canvas.oregonstate.edu/courses/1879154/files/93831999/download?download\\_frd=1](https://canvas.oregonstate.edu/courses/1879154/files/93831999/download?download_frd=1)) for testing the localhost to Atlas connection. The following video shows how to use the app to test our connection to MongoDB.



usefulness of query languages and many now support some form of query language. However, the lack of standardization means that the query languages of different document DBMSs can vary a lot.

### Choosing a DBMS

When developing applications, choosing a type of DBMS and then a specific DBMS within that type of DBMS is an extremely important architectural choice. This choice requires taking into consideration many factors, including the scale and performance needs of the application, maintenance and availability requirements, geographical distribution of the users, as well as other considerations. This is a topic far beyond the scope of this course.

In this course we will use MongoDB which is a document DBMS. The format of documents in MongoDB closely resembles JSON. More specifically, MongoDB internally stores documents in a format called BSON, which is a binary form of JSON. But the MongoDB API exposes the documents as JSON and the JSON format makes it very easy to use MongoDB in Node programs.

### Setting Up MongoDB

We will use MongoDB's Community Edition in this course. One way to use MongoDB is to install and run it on our local machine. Directions for doing this for various operating systems, including Windows and macOS are available at the <https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-windows/>.

Another alternative is to use a hosting service for running MongoDB. MongoDB Atlas, the managed MongoDB service, provides a fully managed MongoDB service. We recommend it, especially on Mac OS.

### MongoDB on Atlas Cloud Cluster

MongoDB Inc., the company behind the MongoDB DBMS services, also have a free offering that you can use to run a server on Atlas Cloud. Follow along with the video to see

### Connecting Our Node App to a MongoDB Server

Let's now test whether we have everything setup correctly for Node apps running on our machine to connect to a MongoDB server. Numerous npm packages are available for use in Node.js apps to connect to a MongoDB server. We will use the npm package `Mongoose` in this course. Here is a [link](#) to a small app that we will use to test the connection to the MongoDB server on Atlas Cloud.

## Summary

In this exploration, we gave a quick introduction to DBMSs. We discussed two types of DBMSs, relational DBMS, and document DBMS. We then looked at MongoDB, which is a popular document DBMS.

## Additional Resources

Here are some references to learn more about the topics we discussed in this exploration.

- There are numerous textbooks that cover DBMS in-depth. **Database System Concepts** by Avi Silberschatz, Henry F. Korth, and S. Sudarshan is one popular and comprehensive textbook.
- The short book **NoSQL Distilled** by Pramod J. Sadalage and Martin Fowler gives an excellent distillation of concepts underlying many types of non-relational DBMSs. But the book requires a deeper understanding of DBMS than is provided in this course. Thus, it is likely to be more useful to you after you have taken the course CS 340 "Introduction to Databases."
- The Wikipedia article on [Document-oriented Databases](https://en.wikipedia.org/wiki/Document-oriented_database) ([https://en.wikipedia.org/wiki/Document-oriented\\_database](https://en.wikipedia.org/wiki/Document-oriented_database)) gives a good, but brief, introduction to Document DBMSs.