Node.js Interface with Database

What is a database

A database is a piece of software used to store data.

Data can be any piece of information we may want to use at a later time.

Common databases: MySQL, SQL Server, Oracle, Db2, PostgreSQL, **MongoDB**, Redis, Cassandra, SQLite, Couchbase, etc.

Database enable the application to "remember" information.

User => application => API or backend=>data layer=> database=>collections or tables.

Compare SQL to NoSQL

SQL Pros

- Data is nicely organized in appropriate tables, which reduces redundant information.
- The split structure allows use to join data in any way. Numerous join types are available and can be done with any number of tables.
- Data model requires some thought, This ensures data is consistent and easy to work with.
- Prevent developers from sloppily entering data with the intent to organize it later.

NoSQL Pros

- Data can easily be nested, allowing everything related to a particular entity to be in one sport.
- With nested documents, we don't have to worry about joining documents in most situations.
- Data model is flexible, allowing for easy iterations through the development process and easy-of-mind if the structure needs to change.
- The flexibility of MongoDB removes the headache of focusing so much on the database, Developers are free to focus on coding.

Setup a MongoDB Atlas Cluster

- https://account.mongodb.com/account/login

To sign up
then we can access
https://cloud.mongodb.com/
New Project
Create a Cluster

Select M0 Free with storage 512MB,
Set your cluster name "myCluster"
Provider we can select AWS, us-east
create database user
Security/Database Access to add new database user
Security/Network to add IP address

Connect with MongoDB Compass

- Database / Connect/Compass

  Download MongoDB Compass
  Copy the connection string, then open MongoDB Compass

  Open MongoDB Compass
  Databases/Create Database to create new database with collection
  Select database/collection AddData/Insert Document to insert data

Use MongoDB with Node.js

- MongoDB is a non-relational database.
- Data inside MongoDB is stored as JSON objects.
- Easy to integrate MongoDB with Node.js

Use MongoDB with Node.js

Npm install mongodb

```
import { MongoClient, ServerApiVersion} from 'mongodb';
const client = new MongoClient(uri, {
    serverApi: {
      version: ServerApiVersion.v1,
      strict: true,
      deprecationErrors: true,
    }
  });
  await client.connect();
  const dbRecords = await  client.db(DBNAME).collection(COLLECTION_NAME).find({});
   InsertOne/replaceOne/deleteOne
```

Use mongoose schema

- Npm install mongoose
- Define mongoose model schema

const myModel = mongoose.model(COLLECTION_NAME, schema);

const conn = await mongoose.connect(uri)
myModel.find(id)

myModel.create(record)

Use SQL Database with Node.js

- Install MySQL
- Install MySQL workbench
- Use workbench to create database, table, insert records into table.
- Npm install mysql

```
connection = mysql.createConnection(host, database, user, password)

connection.connect()
connection.query(sqlQuery, (err, data)=>{})

connection.end()
```