CoGrammar

Welcome to this session:
Databases and the MongoDB database

The session will start shortly...

Questions? Drop them in the chat.
We'll have dedicated moderators answering questions.

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Skills Bootcamp Cloud Web Development

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Skills Bootcamp Cloud Web Development

- For all **non-academic questions**, please submit a query: ***www.hyperiondev.com/support***
- **Report a safeguarding incident: *www.hyperiondev.com/safeguardreporting***
- We would love your feedback on lectures: ***Feedback on Lectures***
- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

Learning Outcomes

- **Differentiate** between relational and NoSQL databases.
- **Describe** the features of MongoDB and its role in full-stack development.
- **Set up and interact** with a MongoDB database using the Mongo shell and MongoDB Atlas.
- **Implement** secure database practices, including IP whitelisting and user management.

Lecture Overview

- Introduction to Databases
- Relational Databases
- NoSQL Databases
- MongoDB

What is the primary purpose of a database?

A. To store and manipulate data
B. To connect front-end and back-end systems
C. To serve as an interface for users
D. To process payments

Which of the following is a feature of NoSQL databases?

A. High scalability
B. Fixed schema
C. High transaction consistency
D. Relational table structures

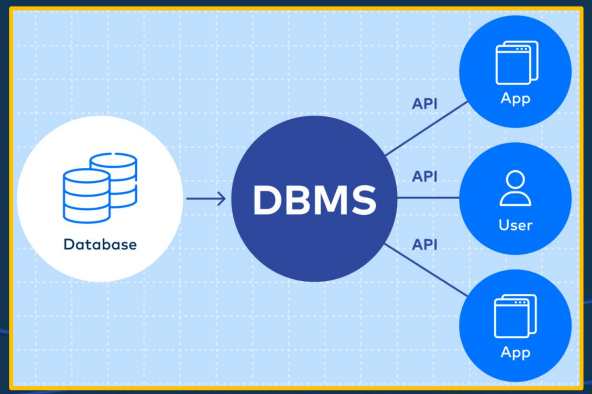
Databases

A large container of data with the ability to order the data in multiple ways, while providing access to the data itself.

- **Data** refers to **raw, unprocessed** facts. Once data has been processed, we call it **information**.
- The production of accurate, timely and relevant information is the key to good **decision-making**, which is the key to a **business' survival** in a competitive global environment.
- Timely and useful information requires accurate data, which must be captured properly and stored in a format that is easy to access and process

DBMS

- A database is usually controlled by a database engine, commonly known as a **Database Management System (DBMS).**
- DBMSs serve as a **tool** between a user and their data, **organising** and **cataloging** the data for **quick and easy retrieval.**
- The data and the DBMS, and the applications associated with them are referred to as a **database system**, usually shortened to **database.**



CoGrammar

DBMS

- The **advantages** of the DBMS are:
  - **Data sharing:** Better access to more, better managed data across applications and users.
  - **Data integration:** Unified view of well-managed data combined from multiple sources.
  - **Data consistency:** Minimised risk of different versions of the same data stored in different places.
  - **Data access:** The DBMS makes it possible to produce quick answers to spur-of-the-moment requests for data.

Types of Databases

Single/Multi-user Database	Refers to how many users can work on the database at the same time.
Enterprise Database	A multi-user database that supports more than 50 users and an entire organisation, across departments.
Centralised/Distributed Database	Refers to how many sites the database is distributed across.
Structured/Unstructured Database	Refers to whether data is stored in the form collected in or if it has been processed to facilitate operations.

Let’s Breathe!

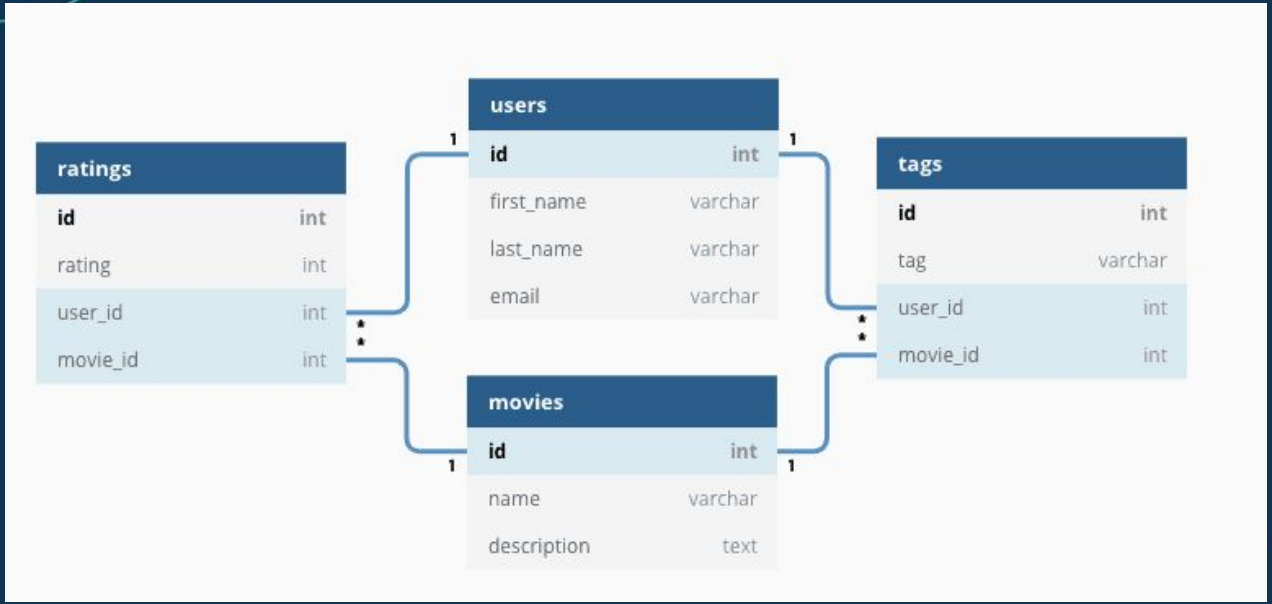
Let’s take a small break before moving on to the next topic.

CoGrammar

Relational Databases

Any database system that allows data to be associated and grouped by common attributes.

- Relational databases are comprised of a number of tables (**relations**), within each are:
  - Rows also known as records or tuples
  - Columns also known as attributes or fields
- Each record is identified with a **unique key**, known as the **primary key**.
- Records from one table can be references in other tables using their key, in this case they are called **foreign keys**.
- Each table/relation represents one **“entity type”**.

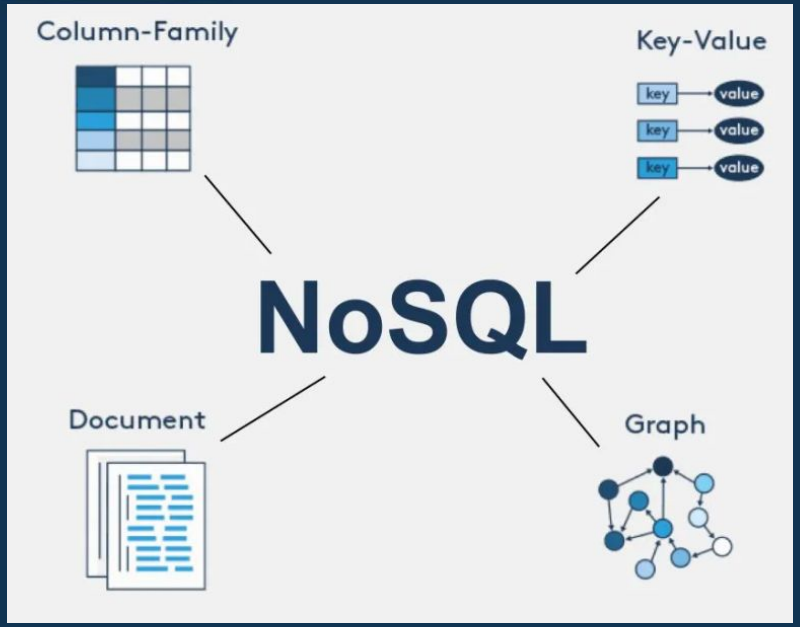


Source: [What is a Relational Database? Definition and FAQs | HEAVY.AI]

CoGrammar

NoSQL Databases

- The performance of relational databases degrades as the volume of data increases.
- Web applications usually have to store massive amounts of data, so NoSQL databases were developed to improve performance.
- NoSQL databases have the following characteristics:
  - Not based on the relational model.
  - Support distributed database architectures.
  - High scalability, high availability and fault tolerance.
  - Support large amounts of sparse data.
  - Geared toward performance rather than transactional consistency



Source: NoSQL Databases: Empowering Modern Data Management

Types of NoSQL DBs

Key-value store databases	Simplest form of the NoSQL DB. Every item is stored as a key and a value.
Column-oriented databases	A key is used to identify values but can identify multiple values instead of one.
Document-store databases	A key is used to identify a particular document (like XM, JSON, PDF, etc.)
Graph databases	Graph structure (nodes connected by links or edges) is used to store data.
Object-oriented databases	Combines OOP and database principles.

MongoDB

A document store and NoSQL database, made up of collections and documents.

- Collections: A group of documents, similar to an entity or table in RDBs.
- Documents: Equivalent to a record in an RDB (or row in a RDB table).
- MongoDB uses **Binary JSON** (BSON) which uses JSON files and stores **type** information, which makes it **quicker** and **more efficient** to use.
- If a user wants to access, add, or change any information that needs to persist, they will need access to the MongoDB database.
- Clients interact with a web server that runs Node.js, which makes use of **MongoDB drivers** to communicate with MongoDB.

Installation

Installing MongoDB to use Mongo and Atlas to host MongoDB on the cloud.

1. Install MongoDB's free Community Server.
2. Configure MongoDB Atlas:
    a. Enter your information here.
    b. On the Database Deployments page, click of Build a Database.
    c. Under 'Cloud provider and Region', select AWS and any free tier region.
    d. Under 'Cluster Tier', select the free M0 option.
    e. You can rename your cluster under 'Cluster Name'.
    f. Click 'Create' to create your cluster.
    g. Get the connection string to connect to the database server.

Shell Commands

- `show dbs;`
  - List all the databases in your cluster.
- `use db_name;`
  - Select a database or create it if it does not exist.
- `show collections;`
  - Shows all the collections in the previously selected database.
- `db.dropDatabase();`
  - Deletes the selected database.

Mongoose

A library that makes working with the MongoDB driver simpler.

1. Install Mongoose using NPM:
    a. `npm install mongoose`
2. Create a schema which outlines the data in our database and how it is organised and structured.
3. Create a controller file to perform data manipulation.
4. Connect to the database and execute operations.

CRUD Operations

Create, Read, Update and Delete

- These are the 4 basic operations which act as the **foundation** of any computer programming language.
- We need to understand CRUD in Mongoose to interact with databases.

1. **Create:** To add or insert collections or documents into it.
   a. `insertOne({document});`
   b. `insertMany([{document1}, {document2}]);`
2. **Read:** To retrieve or fetch documents from your collection.
   a. `find()`

CRUD Operations

3. Update: To modify documents within a collection.

a. `updateOne({field}, { $set: {new_document}});`

b. `updateMany({field}, { $set: {new_document}});`

4. Delete: To remove or delete documents from a collection.

a. `deleteOne({field});`

b. `deleteMany({field});`

Which MongoDB feature improves scalability for large applications?

A. Fixed schema
B. Distributed architecture
C. Single-user support
D. Static data storage

What command creates a new database in the Mongo shell?

A. create database [name]
B. use [name]
C. init db [name]
D. db.createDatabase([name])

CoGrammar

Q & A SECTION

Please use this time to ask any questions relating to the topic, should you have any.

Thank you for attending

CoGrammar

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education