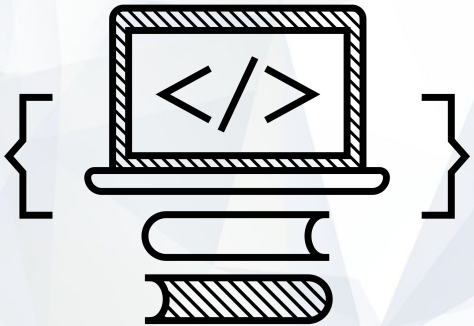
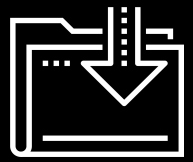


Object-Relational Mapping (ORM)

Coding Boot Camp

Module 13



What are some of the challenges of using plain SQL in JavaScript?

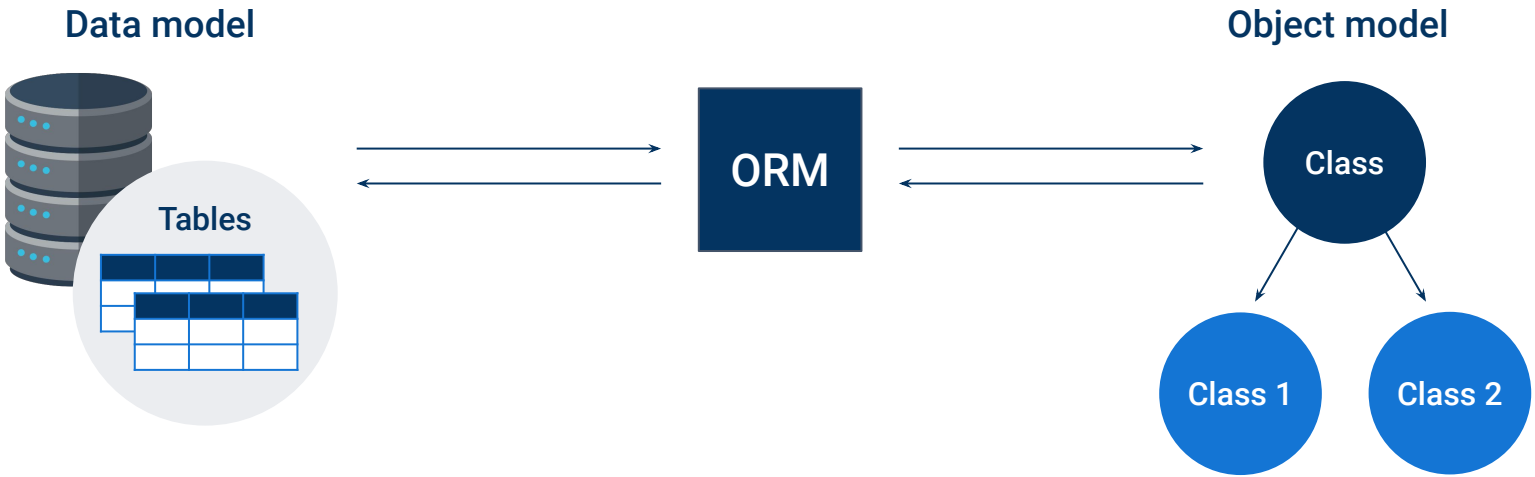
Challenges of SQL in JavaScript

- It is prone to accidental syntax errors.
- Complicated queries can be hard to follow.
- It requires extra work to validate and secure data.
- Similar routes can lead to repetitive queries being written.
- Data relationships aren’t obvious just by looking at the code.

What is object-oriented programming?

Object-Oriented Programming

Object-oriented programming is a programming paradigm based on the concept of "objects", which can contain data and code: data in the form of fields, and code, in the form of procedures.



Object-Oriented Programming

01

Code is organized using objects instead of functions.

02

Objects can inherit properties and methods from other objects.

03

Multiple objects can be created from the same blueprint classes or constructor functions.

How can objects help us manage SQL queries in JavaScript?

Objects and SQL

01

We can set up object methods to run generic SQL queries based on the method's parameters.

02

We can structure objects to mimic how the data is stored in the database, eliminating ambiguity.

03

We can import objects into any other module that needs to execute its SQL queries.

Is there a library that might already do this for us?

Sequelize is an object-relational mapper (ORM) that can be installed with npm.

Sequelize

Sequelize provides the following benefits:

- Allows you to model your data as JavaScript classes.
- Abstracts SQL queries to simpler object methods.
- Provides built-in validation checks for securing data.
- Makes it easier to visualize and join relational data.
- And more!

How can we learn to use and implement Sequelize?

Sequelize and other ORMs were created to make managing and using data easier, but they still come with a learning curve!

How to Learn Sequelize

You can try the following strategies to learn Sequelize:

- Read the official documentation and practice with the provided examples.
- Reverse-engineer finished code to see how something was accomplished.
- Build something from scratch.
- Debug a broken app.
- And most importantly, ask questions!

Instructor Demonstration

Mini-Project