BÁO CÁO BÀI TẬP 1

SỬ DỤNG ORM SEQUELIZE TRONG LẬP TRÌNH BACK-END KẾT HỢP VỚI NODEJS, EXPRESS VÀ MYSQL

# 

Thao tác với dữ liệu trong cơ sở dữ liệu là một việc không thể thiếu với bất cứ lập trình viên Backend nào . Tuy nhiên có những người không được học về các loại cơ sở dữ liệu, họ học không tốt, đặc biệt trong các CSDL SQL . Nếu họ không nắm chắc các câu lệnh truy vấn thì việc truy xuất các dữ liệu trong các bảng của database gần như là không thể. Chính vì những lý do đó, nhằm hỗ trợ lập trình viên trong việc thao tác dễ dàng với database mà không cần phải hiểu quá rõ về những loại database mình đang dùng , rất nhiều ORM được phát triển và ngày càng trở thành 1 phần không thể thiếu đối với các lập trình viên. Cùng với việc ngôn ngữ Nodejs là ngôn ngữ khá được nhiều người tin dùng thì ORM Sequelize đã được phát triển để trở thành tệp đính kèm không thể thiếu mỗi khi chúng ta dùng Nodejs.

# MỤC LỤC

# ORM là gì ? Sequelize là gì ?

# Cài đặt Sequelize

# Kết nối với database

# Tạo bảng trong database bằng Sequelize

# Truy vấn cơ bản

# Thực hiện truy vấn liên kết nhiều bảng trong database

# ORM là gì ? Sequelize là gì ?

# ORM là gì ?

* ORM (Object Relational Mapping), là một kỹ thuật/cơ chế lập trình thực hiện ánh xạ CSDL sang các đối tượng trong các ngôn ngữ lập trình hướng đối tượng như Java, C# …Việc sử dụng ORM cho phép lập trình viên thao tác với database một cách hoàn toàn tự nhiên, dễ hiểu thông qua các đối tượng. - - Với ORM lập trình viên không hiểu rõ về loại database, kiểu dữ liệu trong database…

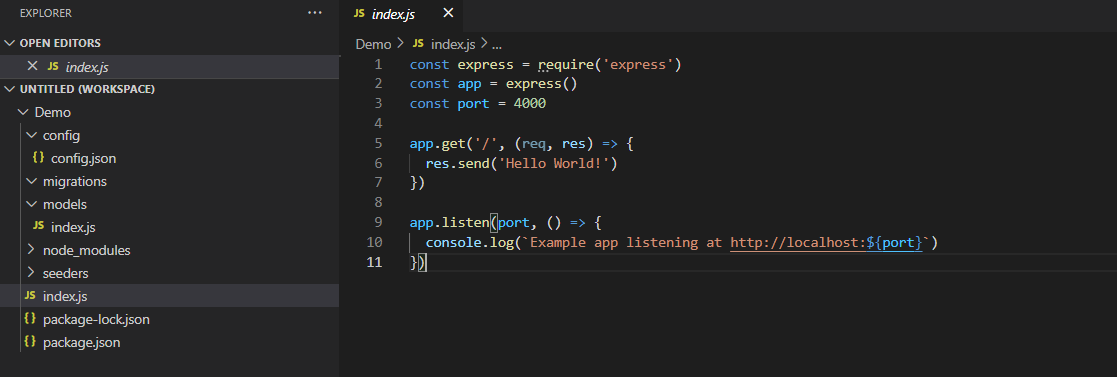
# Sequelize là gì ?

* Sequelize là một ORM cho Nodejs
* Sequelize nó giống như một framework cho database.Khi thực hiện truy vấn với database thay vì chúng ta truy vấn database với các câu query thuần thì với Sequelize , chúng ta sẽ sử dụng các hàm do Sequelize cung cấp để truy vấn DB. Bên cạnh đó, Sequelize còn cho phép người dùng định nghĩa cấu trúc table, validate, v.v..
* Sequelize nó hỗ trợ các database như PostgreSQL, MySQL, MariaDB, SQLite và MSSQ

# Cài đặt Sequelize

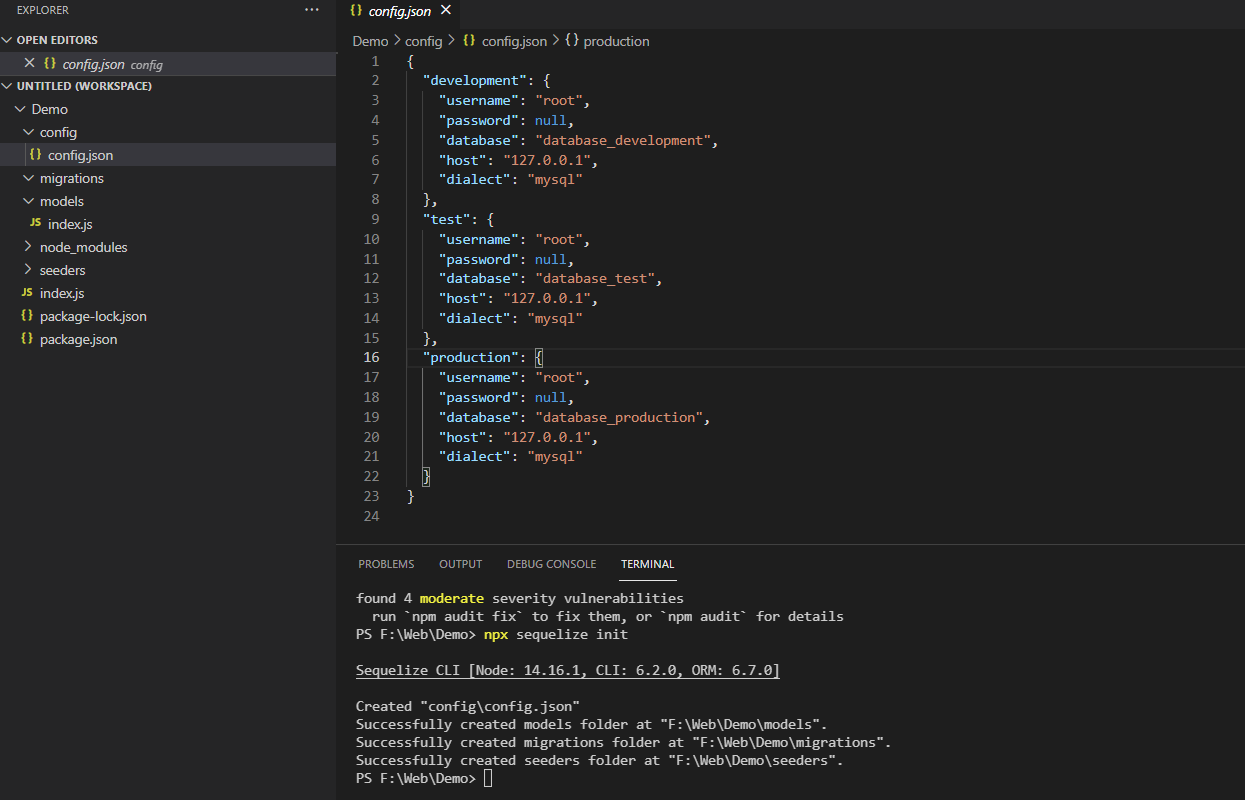
1. Yêu cầu trước khi cài đặt Sequelize

* Khởi tạo project bằng npm init
* Tạo 1 file index.js để chạy server với localhost , nội dung file có thể lấy từ Getting Started -> HelloWorld trên Expressjs



b. Các bước cài đặt

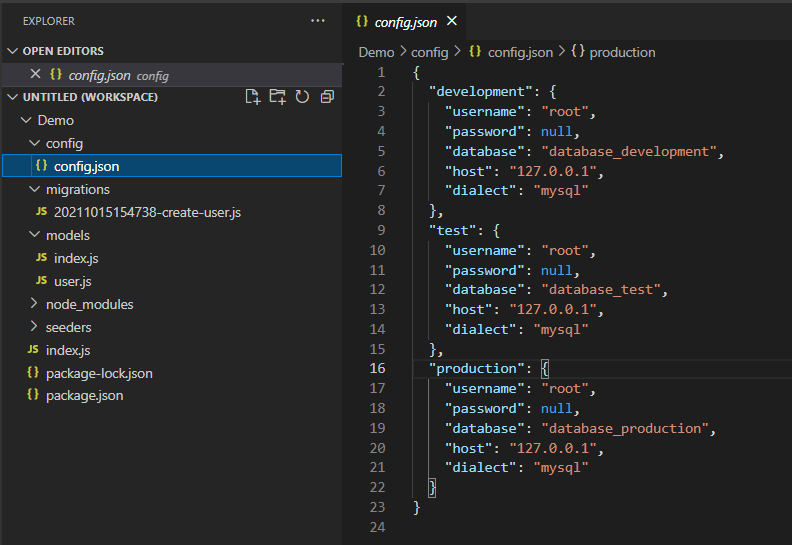
* Cài đặt sequelize bằng câu lệnh : **npm install sequelize**
* Sau đó cài đặt **mysql2** để hỗ trợ cho việc dùng sequelize: **npm install mysql2**
* Sau khi đã cài đặt xong 2 modules xong , ta chạy khởi tạo như sau “**npx sequelize init**”
* Đây là kết quả sau khi khởi tạo sequelize



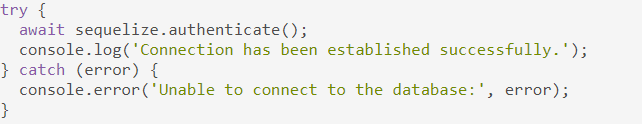
* Các thư mục được tạo ra :
* config: chứa file config.json, là nơi để chúng ta cài đặt kết nối với database
* models: Chứa các models của project, với từng loại kiến trúc code mà có thể là nơi dùng để xác định, cấu hình các thành phần trong 1 bảng nào đó trong database
* migrations: Chứa tất cả các tệp có chức năng thực hiện tạo các bảng đã được cấu hình trong models vào database
* seeders: Chứa tất cả các tệp dành cho việc khởi tạo master data,tạo các giá trị trong bảng sẵn khi mà bảng được tạo.

# Kết nối với database

* Kết nối với database trong file config.json()



* Khi khởi tạo bằng Sequelize , module đã tạo sẵn cho người dùng 3 objects để người dùng có thể sử dụng cho việc dev, test cũng như hoàn thành đưa ra thành sản phẩm chính thức.
* Kiểm tra việc connect có thành công hay không ?
* Ta sử dụng hàm .authenticate() để kiểm tra connect.
* Ta sử dụng try-catch để kiểm tra, nếu connect với database thành công thì sẽ hiện ra dòng “ Connect has been established successfully”
* Ngược lại sẽ hiện ra dòng bên dưới cũng như kèm theo lỗi



* Ta chèn đoạn code này vào sau vào hàm listen để sau khi chương trình chạy localhost tại 1 port mặc định trước, nếu thành công rồi thì mới kiểm tra tới việc kết nối có thành công hay không của database.

# Tạo bảng trong database bằng Sequelize

1. Model trong Sequelize

* Model là bản chất của Sequelize, mỗi một model đại diện cho 1 bảng trong cơ sở dữ liệu.
* Model chưa các thông tin như tên bảng trong database, tên các cột(trường), và kiểu dữ liệu của chúng.
* Mô hình được cấu hình theo 2 cách trong: sequelize.define và Extending Model

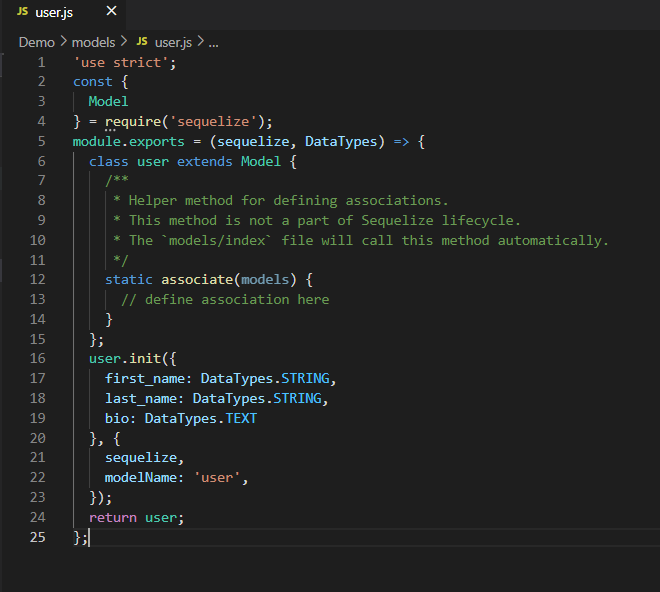
1. Sử dụng Migrations và Extending Model

* Để tạo Extending Model : ta sử dụng câu lệnh sau:

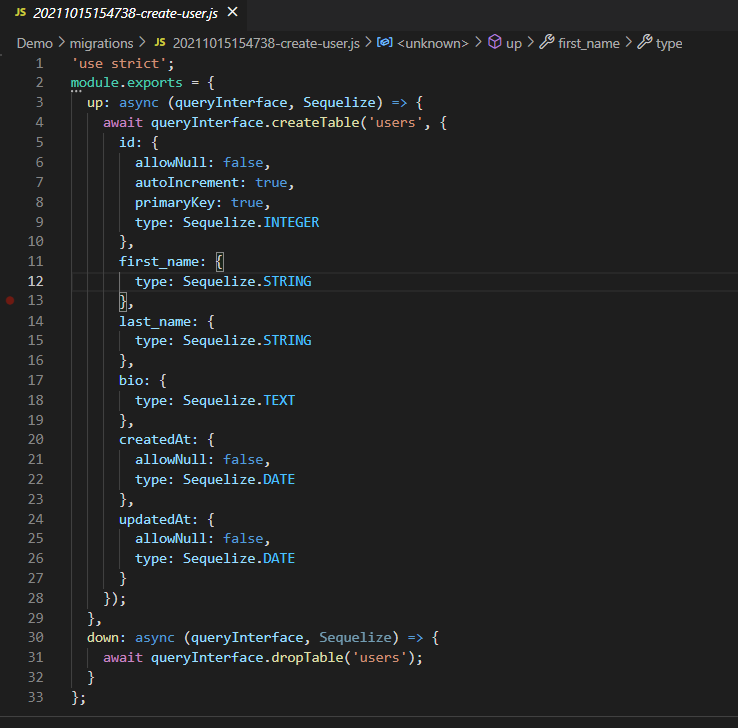
npx sequelize model:create --name [ten\_model] --attributes [ten\_attr]:[datatype\_attr],...

* Trong ví dụ này, ta sử dụng câu lệnh sau :

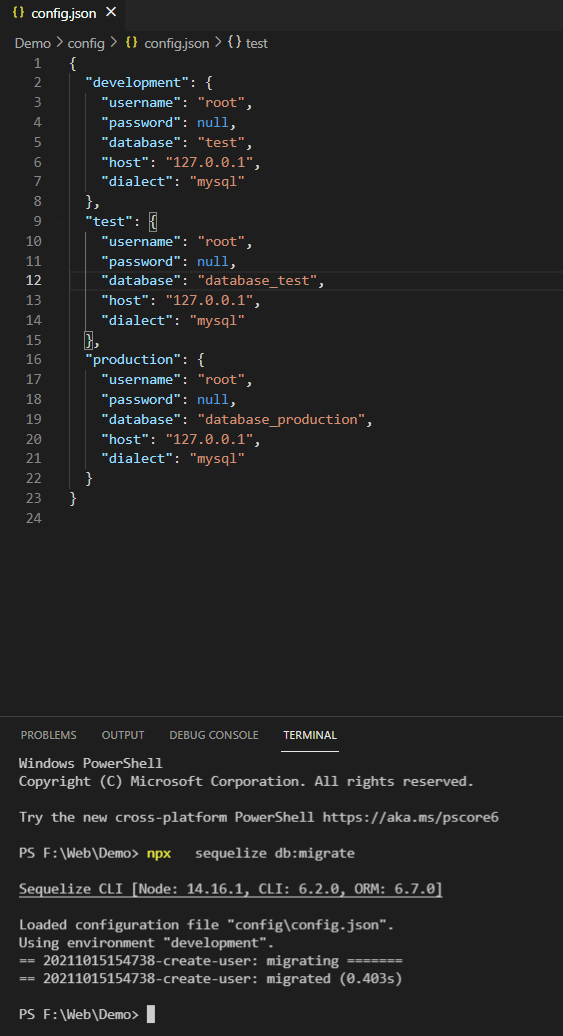
npx sequelize model:create --name user --attributes first\_name:string,last\_name:string,bio:text



* Đồng thời, một file trong tệp migrations được tạo ra với hình ảnh như sau:

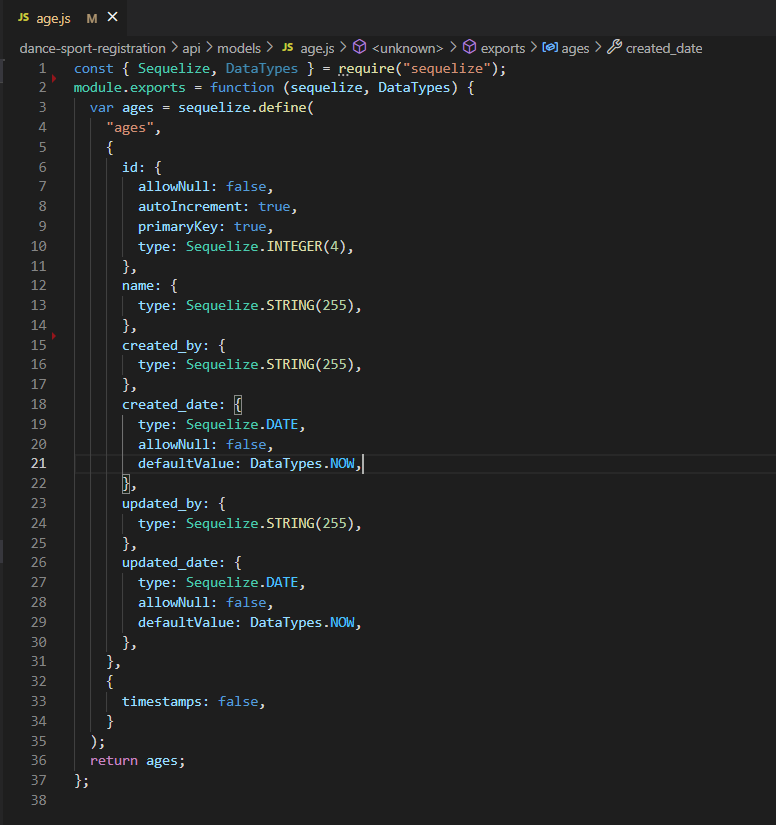


* Cuối cùng, để thực hiện việc tạo bảng theo model được cấu hình sẵn thì ta sử dụng câu lệnh sau :

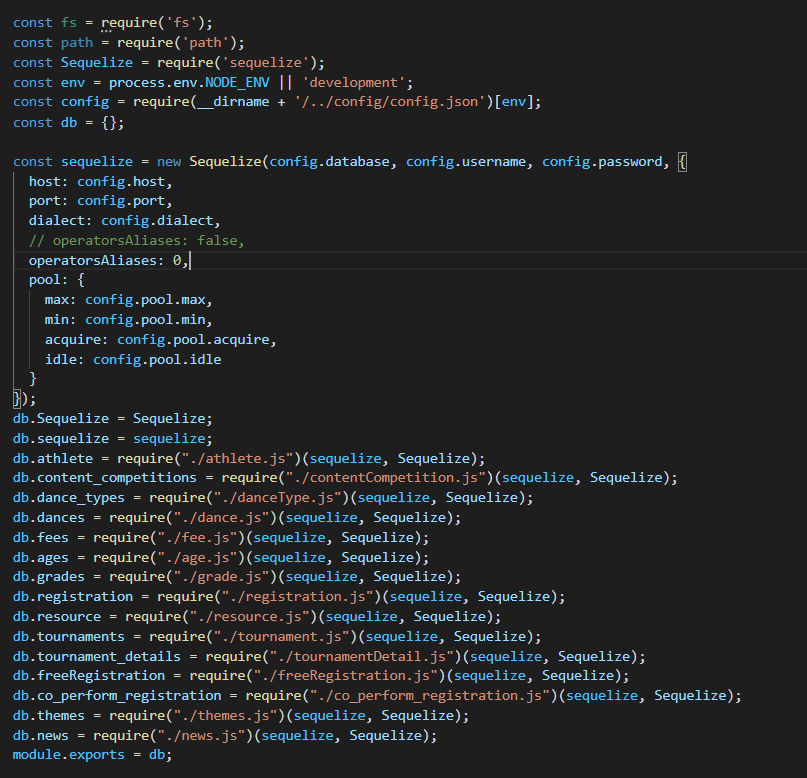


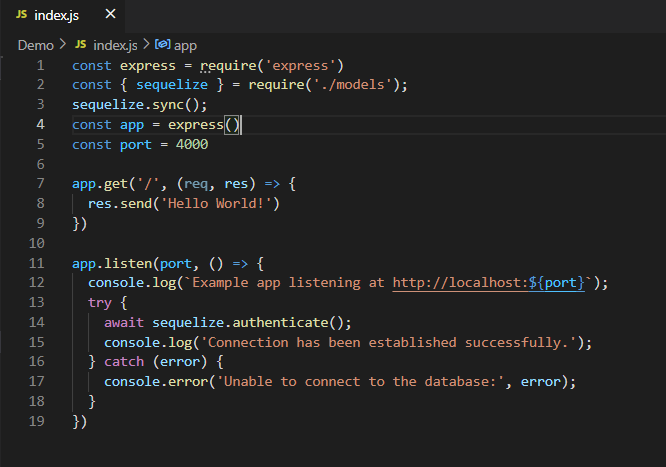
1. Sử dụng tạo model bằng sequelize.define

* Cách này thì chúng ta sẽ tốn thời gian hơn nhưng code của chúng ta cũng sẽ gọn hơn, tất cả đều nằm trong một file model duy nhất.



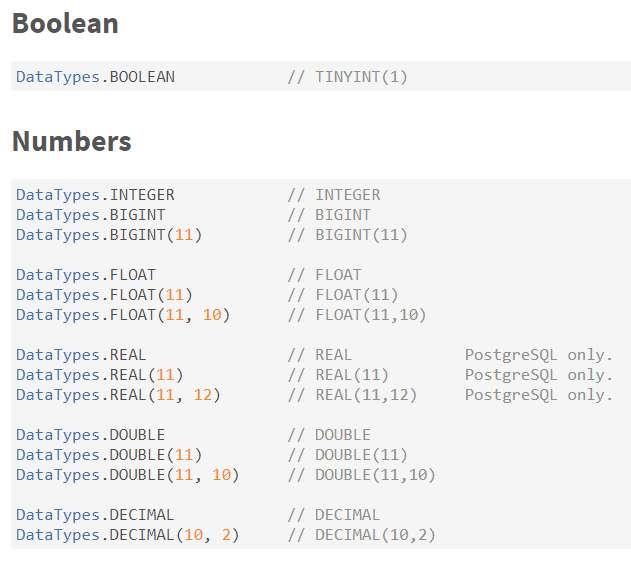
* Tuy nhiên để thực hiện tạo bảng trong database thì ta sẽ dùng hàm đồng bộ để thực hiện việc đó.
* .sync(): Sẽ tạo ra bảng đó nếu bảng đó chưa tồn tại
* .sync({force: true}): Sẽ tạo ra bảng đó bằng bất cứ giá nào, nếu nó đã tồn tại thì sẽ xóa đi rồi tạo lại
* .sync({alter: true}): Sẽ chỉ thay đổi những cột được thay đổi trong model mà ko xóa đi những cột khác.
* Hoặc ta cũng có thể dùng sequelize.sync() để tự động đồng bộ các bảng theo những thay đổi của model

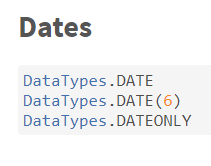




1. Các kiểu dữ liệu có trong Sequelize

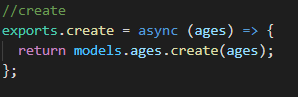




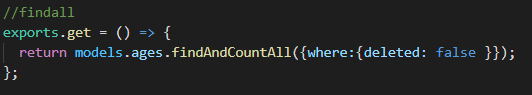


# Truy vấn cơ bản

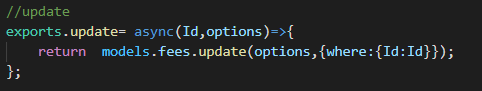
* Để thêm các hàng vào trong cột thì ta dùng hàm create.



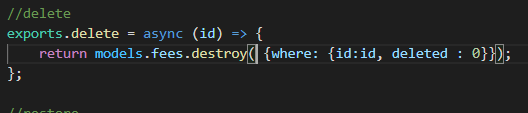
* Để tìm các hàng trong cột thì ta dùng findAll ,findAndCountAll,findOne hoặc findByPk.



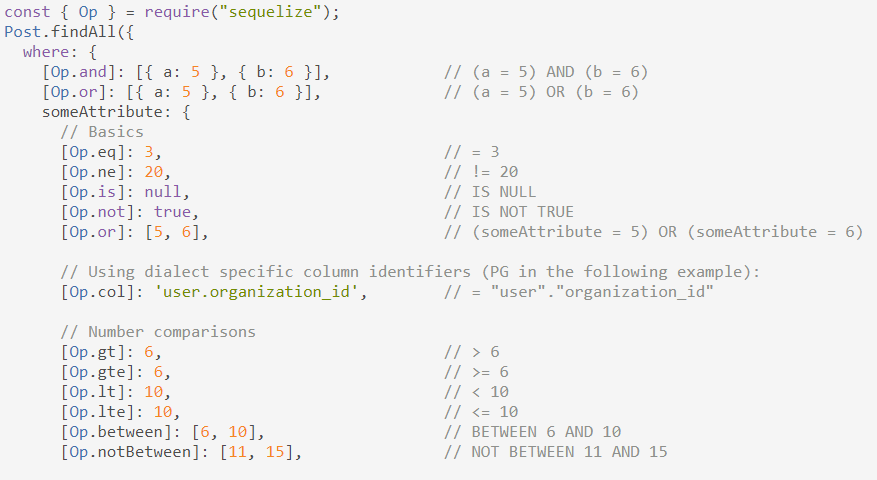
* Để chỉnh sửa các bản ghi thì ta sử dụng hàm update.



* Để xóa các bản ghi ta sử dụng hàm destroy.



* Sử dụng mệnh đề Where để lọc truy vấn

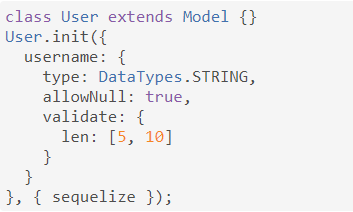




* Một số validate

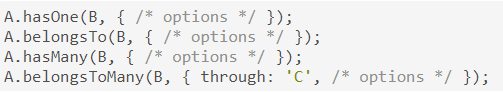


Ví dụ về validate được đặt trong model



1. Thực hiện truy vấn liên kết nhiều bảng
2. Liên kết trong Sequelize

* Sequelize hỗ trợ các liên kết tiêu chuẩn như 1-1, 1-N, N-N.
* Sequelize cung cấp 4 loại liên kết để kết nối
* HasOne: 1-1
* BelongsTo: 1-1
* HasMany: 1-N
* BelongsToMany: N-N



* Liên kết trong Sequelize thường được xác định theo cặp nên:
* Để tạo ra liên kết 1-1 : sử dụng hasOne và belongsTo
* Để tạo liên kết 1-N: sử dụng hasMany và belongsTo
* Để tạo liên kết N-N: sử dụng 1 cặp belongsToMany

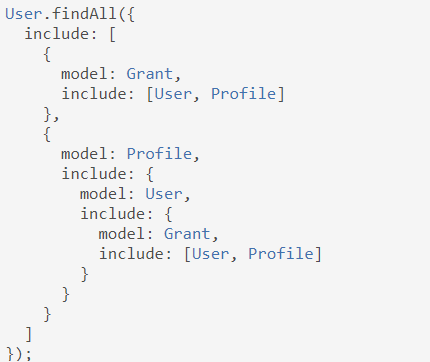
1. Tạo liên kết để thực hiện truy vấn

* Có 2 cách: Khai báo trong model hoặc ở bất cứ file nào khi cần dùng đến nó tại file đó
* Cách 1 :

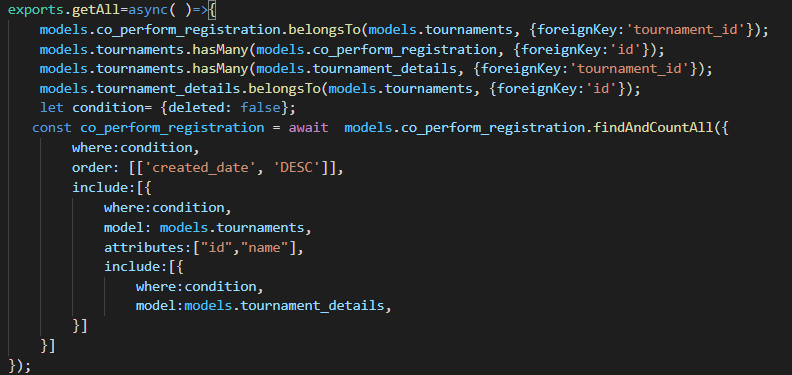


1. Thực hiện truy vấn

* Ta sử dụng cấu trúc sau :



* Ngoài ra , trong các model ta có thể thêm thuộc tính attributes để lấy ra những cột mình mong muốn



KẾT LUẬN :

* Ưu điểm của Sequelize
* Miễn phí,Nhanh, dễ nhớ, thân thiện
* Không phải nhớ các câu truy vấn SQL
* Hỗ trợ trong việc cấu hình kiến trúc code cho project của bạn
* Hỗ trợ được nhiều CSDL SQL
* Là một mã nguồn mở nên việc tiếp cận, tìm hiểu rất dễ dàng
* Sử dụng cấu trúc Promise
* Hơn nữa hỗ trợ rất tốt cho CSDL NoSQL đang rất nổi thời điểm hiện tại là MongooseDB
* Nhược điểm
* Để sử dụng được phải cần một thời gian tìm hiểu lâu vì nội dung kiến thức lớn
* Việc truy vấn sẽ chậm hơn so với truy vấn trực tiếp bằng câu lệnh truy vấn SQL
* Việc truy vấn có liên kết giữa các bảng còn khá rối, dễ nhầm lẫn

G. Tài liệu tham khảo :

* Cách cài đặt Sequelize: <https://sequelize.org/master/manual/getting-started>
* Cài đặt kết nối với database:

<https://sequelize.org/master/manual/getting-started.html#connecting-to-a-database>

* Tạo model với sequelize:

<https://sequelize.org/master/manual/model-basics.html>

* Các kiểu dữ liệu trong Sequelize

<https://sequelize.org/master/manual/model-basics.html#data-types>

# Truy vấn cơ bản:

<https://sequelize.org/master/manual/model-querying-basics.html>

<https://sequelize.org/master/manual/model-querying-finders.html>

* Sử dụng mệnh đề Where để lọc truy vấn, các validate:

<https://sequelize.org/master/manual/model-querying-basics.html#applying-where-clauses>

<https://sequelize.org/master/manual/validations-and-constraints.html>

* Truy vấn nhiều bảng bằng liên kết:

https://sequelize.org/master/manual/assocs.html

# 