



# SEQUELIZE

# Plan

Introduction

I- Les modèles

II- Les promesses

III- Les fonctionnalités

IV- Démonstration

Conclusion



# Présentation

Sequelize est un ORM basé sur les promesses pour Node.js et io.js

Compatibilité avec PostgreSQL, MySQL, MariaDB, SQLite et MSSQL

## Installation avec NPM :

Installation des drivers “postgres” et d’une dépendance pour (dé)sérialiser du json, installation de Sequelize

```
$ npm install --save-dev pg-hstore pg
```

```
$ npm install --save sequelize
```

```
1.  var sequelize = new Sequelize('database', 'username',  
2.      'password', {  
3.          host: 'localhost',  
4.          dialect: 'mysql' | 'mariadb' | 'sqlite' | 'postgres' | 'mssql',  
5.          pool: {  
6.              max: 5,  
7.              min: 0,  
8.              idle: 10000  
9.          },  
10.         // SQLite only  
11.         storage: 'path/to/database.sqlite'  
12.     });  
13.  
14.  var sequelize = new  
15.  Sequelize('postgres://user:pass@example.com:5432/dbname');
```

# Les modèles - En théorie

Objets permettant d'accéder à une table.

Disposent des noms de colonnes de la table.

Création des attributs `createAt` et `updatedAt` automatique.

Accesseurs pour accéder aux données du modèle.

# Les modèles - En javascript

```
1.  var Modele = db.define('maTable', {
2.      attribut: {
3.          type: Sequelize.STRING,
4.          primaryKey: true,
5.          set: function(val) {
6.              // setter qui mets en majuscule avant ajout dans le modèle
              this.setDataValue('attribut', val.toUpperCase());
          }
7.      }
8.      // Sequelize.STRING, Sequelize.Date ...
9.  })
```

Possibilité d'ajouter des options aux attributs (primaryKey, unique, ...).

Plusieurs types de données disponibles comme INTEGER, STRING, TEXT.

# Les promesses

Système de gestion des méthodes asynchrones

`.then(function() {});`      $\Rightarrow$  exécution de la fonction lorsque la promesse est finie.

`.catch(function() {});`      $\Rightarrow$  exécution de la fonction lors d'une erreur.

`.done(function() {});`      $\Rightarrow$  exécution de la fonction si la promesse fini avec succès.

`.all([promese1, promese2]).then(function() {} );`

$\Rightarrow$  exécution de la fonction lorsque toutes les promesses sont terminées.

# Les fonctionnalités - La syntaxe

```
1.  maTable.sync({force: true})           // Synchronisation du modèle avec la table du SGBD
2.  // Création de la table si elle n'existe pas, force : efface et recrée la table
3.
4.  // Insère une ligne dans la table
5.  maTable.create( {id: "6",  nom: "toto", mail: "toto@univ-lyon.fr"} );
6.
7.  maTable.drop();                       // Efface la table
8.
9.  maTable.findAll({                     // Lance un Select sur la table
10.     attributes: ['id', 'titre', 'description', 'date', 'numC'],
11.     where: {
12.         date: {
13.             gt: 2010                     // Si l'attribut est plus grand que 10
14.         },
15.     }
16. }).then(function(resultat) {
17.     //traitement du résultat
18. });
```

# Les fonctionnalités - Quelques méthodes utiles

Relier une clé étrangère avec une clé primaire d'une autre table

```
1. User.belongsTo(Company);           // Clé étrangère : de un à un
2. Project.hasMany(User, {as: 'Workers'}) // Clé étrangère : de un à plusieurs
```

Exécution d'une requête SQL

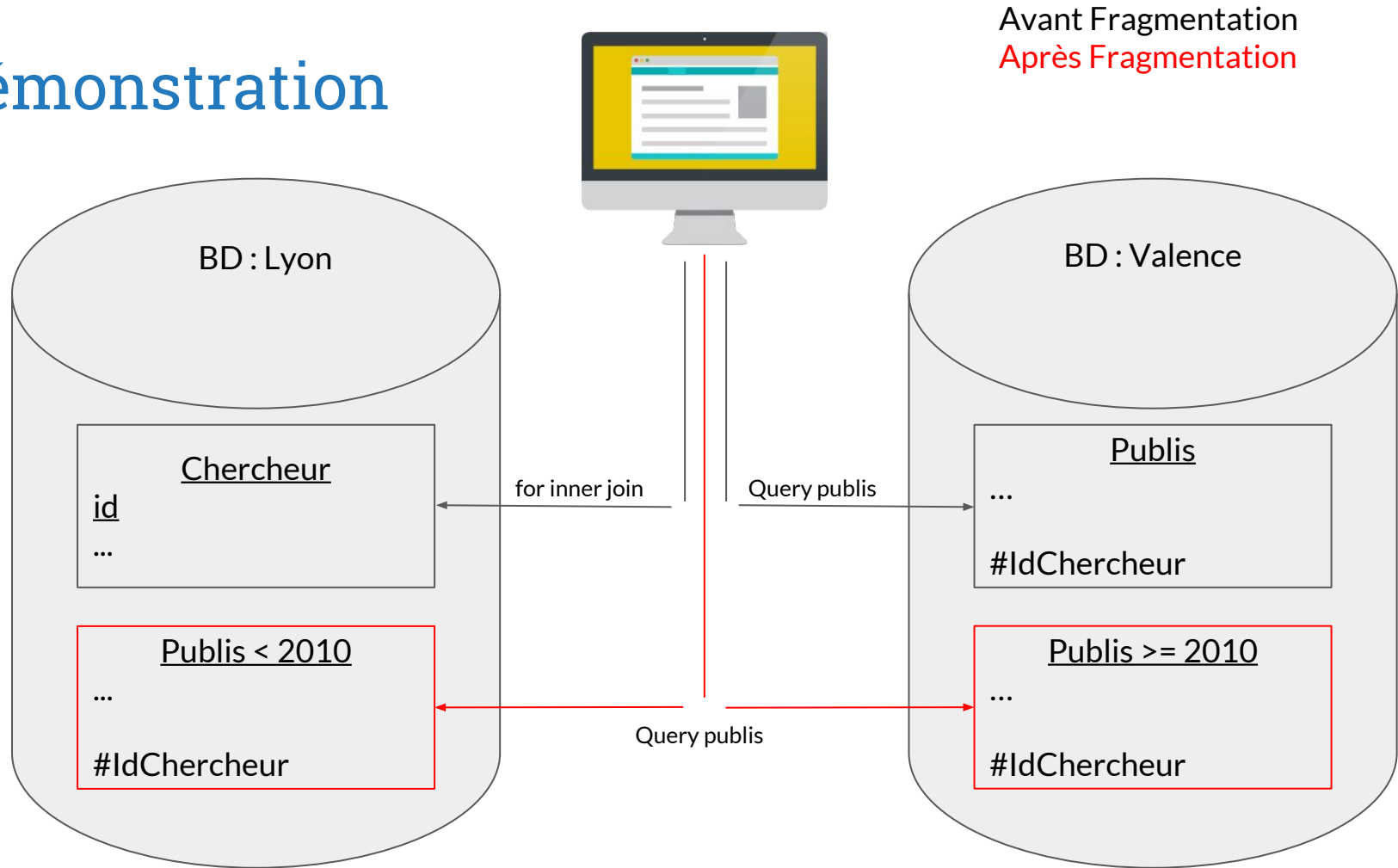
```
1. self.Lyon.query('SELECT * FROM "Chercheurs"', { model: Chercheurs }).then( function(p) {
2.     console.log(p);
3. });
```

Import d'un modèle depuis un fichier stocké sous forme de module

```
1. db.import("./module.js");
```



# Démonstration



# Pour conclure

Manque du SGBD Oracle

Simple à prendre en main

Open Source

Bonne communauté

Autre ORM javascript :

- Bookshelf
- JS-Data.io
- Pg-promise
- offShore
- Firenze.js
- npm.org

Utilisé par :

- timeOff : <http://timeoff.management/>
- Verizon Cloud : <https://www.verizonwireless.com/>
- SEAN : Full stack JS sql based on MEANJS.

# Questions



Documentation officielle : <http://docs.sequelizejs.com/en/latest/>  
Tutoriel : <https://egghead.io/lessons/node-js-sequelize-model-definition-basics>  
Github : <https://github.com/sequelize/sequelize>