Time	~ 1 - 2 hours
Learning Goals	• Speed up a slow app given a huge database

Welcome to Score Board! Things would be great if things weren't so slow.

We have a pre-built Rails application with two models: User and Point. After seeding the database with data, you'll have 100,000 rows in the users table and 1,500,000 rows in the points table. The index page is sl-o-o-ow and it's your job to speed it up without using any fancy technologies like memcache, redis, or Rails' fragment caching.

Start Challenge

1. **Fork** the repository from: https://github.com/NextAcademy/application-speed-matters_2
2. **Clone** the repository into your local machine.

Setting Up the Application

1 Click Start Challenge

2 Get the application running as follows:

```
$ git clone <repo-link>

$ cd application-speed-matters_2

$ bundle install

$ bundle exec rake db:setup
```

Thebundle exec rake db:setup command could take a few minutes. Once it's done run bundle exec rails server and visit http://localhost:3000.

We use a gem called activerecord-import to mass import data into the database. Here is an issue to be aware of: this issue. It should take a few seconds to load. If you look at the last line of your server logs, you'll see it telling you how much time was spent rendering the views versus how much time was executing ActiveRecord methods.

```
Completed 200 OK in 13476ms (Views: 1832.7ms | ActiveRecord: 11625.8ms)
```

Ask yourself!

Where's the bottleneck?

Make it fast!

Your goal is to **get the index page to load in under 200ms**. Yes, that's milliseconds. You're permitted to do the following:

1 Add new fields to the users and points tables

2 Add new indexes to the users and points tables

3 Add new class or instance methods to the User or Point models

The test suite should remain green. If you add any new public methods make sure you add appropriate corresponding tests.

You **should not** change the controller or view code.