Time	1 hour
Learning Goals	• Understand Object-Relational Mapping (ORM). • Understand Active Record. • Understand the relationship between SQL and Active Record. • Understand why we use Active Record to implement an ORM instead of writi

In the last version of the challenge we refactored some simple methods into the base `Database::Model` class, but none of those methods contained actual SQL. All student-specific SQL remained in the `Student` class. Let's change that.

Work with your repo from the previous challenge (Fake ActiveRecord: Part 1).

Objectives

Refactor SQL Into the Base Class

Remember, the base class only knows what it's told. Somehow the `Student` and `Cohort` classes will have to get the following information to the base class:

1. What table am I associated with?
2. What is my primary key?
3. What class do I need to instantiate when returning results?

One possible strategy for this is to see how we set `attribute_names`. There are others. Refactor the following methods into `Database::Model`:

1. `Student.all` and `Cohort.all`
2. `Student.create` and `Cohort.create`
3. `Student.where` and `Cohort.where`
4. `Student.find` and `Cohort.find`
5. `Student#new_record?` and `Cohort#new_record?`
6. `Student#insert!` and `Cohort#insert!`
7. `Student#update!` and `Cohort#update!`

Feel free to play around in the ActiveRecord, Jr. console to get a feel for how it works. And remember , once a method is moved to `Database::Model` you might need to change hard-coded references to the classes the method came from.

Submit Your Refactored Code

Copy your ammended files: `student.rb`, `cohort.rb`, and `database_model.rb` into separate files into this new repo (after you've started challenge and cloned the new repo).