

# In this lecture, we will discuss...

## ✧ Active Record **CRUD**

- **Create**
- **Retrieve**

# Create (CRUD)

- ✧ Three ways to **create a record** in the database
  1. Use an **empty constructor** and (ghost) attributes to set the values and then call `save`
  2. Pass a **hash of attributes** into the constructor and then call `save`
  3. Use `create` method with a hash to create an object and save it to the database in **one step**



```
~/fancy_cars$ rails c
Loading development environment (Rails 4.2.3)
irb(main):001:0> Person.column_names
=> ["id", "first_name", "last_name", "created_at", "updated_at"]
irb(main):002:0> p1 = Person.new; p1.first_name = "Joe"; p1.last_name = "Smith"
=> "Smith"
irb(main):003:0> p1.save
  (0.2ms) begin transaction
  SQL (1.1ms) INSERT INTO "people" ("first_name", "last_name", "created_at", "updated_at") VALUES (?, ?, ?, ?) [["first_name",
, "Joe"], ["last_name", "Smith"], ["created_at", "2015-09-08 02:08:10.357211"], ["updated_at", "2015-09-08 02:08:10.357211"]]
  (0.6ms) commit transaction
=> true
irb(main):004:0> p2 = Person.new( first_name: "John", last_name: "Doe"); p2.save
  (0.1ms) begin transaction
  SQL (0.3ms) INSERT INTO "people" ("first_name", "last_name", "created_at", "updated_at") VALUES (?, ?, ?, ?) [["first_name"
, "John"], ["last_name", "Doe"], ["created_at", "2015-09-08 02:09:11.329095"], ["updated_at", "2015-09-08 02:09:11.329095"]]
  (1.6ms) commit transaction
=> true
irb(main):005:0> p3 = Person.create(first_name: "Jane", last_name: "Doe")
  (0.1ms) begin transaction
  SQL (0.3ms) INSERT INTO "people" ("first_name", "last_name", "created_at", "updated_at") VALUES (?, ?, ?, ?) [["first_name"
, "Jane"], ["last_name", "Doe"], ["created_at", "2015-09-08 02:11:18.824233"], ["updated_at", "2015-09-08 02:11:18.824233"]]
  (1.4ms) commit transaction
=> #<Person id: 3, first_name: "Jane", last_name: "Doe", created_at: "2015-09-08 02:11:18", updated_at: "2015-09-08 02:11:18">
```



# Retrieve / Read (*CRUD*)

- ✧ `find(id)` or `find(id1, id2)`
  - Throws a `RecordNotFound` exception if not found
- ✧ `first, last, take, all`
  - Return the results you expect or `nil` if nothing is found
- ✧ `order(:column)` or `order(column: :desc)`
  - Allows ordering of the results. Ascending or descending
- ✧ `pluck`
  - Allows to narrow down which fields are coming back
  - Need to call at the end!

```

irb(main):001:0> Person.all.order(first_name: :desc)
  Person Load (1.5ms) SELECT "people".* FROM "people" ORDER BY "people"."first_name" DESC
=> [#<ActiveRecord::Relation [#<Person id: 2, first_name: "John", last_name: "Doe", created_at: "2015-09-08 02:09:11", updated_at: "2015-09-08 02:09:11">, #<Person id: 1, first_name: "Joe", last_name: "Smith", created_at: "2015-09-08 02:08:10", updated_at: "2015-09-08 02:08:10">, #<Person id: 3, first_name: "Jane", last_name: "Doe", created_at: "2015-09-08 02:11:18", updated_at: "2015-09-08 02:11:18">]>]
irb(main):002:0> Person.all.order(first_name: :desc).to_a
  Person Load (0.2ms) SELECT "people".* FROM "people" ORDER BY "people"."first_name" DESC
=> [#<Person id: 2, first_name: "John", last_name: "Doe", created_at: "2015-09-08 02:09:11", updated_at: "2015-09-08 02:09:11">, #<Person id: 1, first_name: "Joe", last_name: "Smith", created_at: "2015-09-08 02:08:10", updated_at: "2015-09-08 02:08:10">, #<Person id: 3, first_name: "Jane", last_name: "Doe", created_at: "2015-09-08 02:11:18", updated_at: "2015-09-08 02:11:18">]
irb(main):003:0> Person.first
  Person Load (0.2ms) SELECT "people".* FROM "people" ORDER BY "people"."id" ASC LIMIT 1
=> [#<Person id: 1, first_name: "Joe", last_name: "Smith", created_at: "2015-09-08 02:08:10", updated_at: "2015-09-08 02:08:10">]
irb(main):004:0> Person.all.first
  Person Load (0.2ms) SELECT "people".* FROM "people" ORDER BY "people"."id" ASC LIMIT 1
=> [#<Person id: 1, first_name: "Joe", last_name: "Smith", created_at: "2015-09-08 02:08:10", updated_at: "2015-09-08 02:08:10">]
irb(main):005:0> Person.all[0]
  Person Load (0.1ms) SELECT "people".* FROM "people"
=> [#<Person id: 1, first_name: "Joe", last_name: "Smith", created_at: "2015-09-08 02:08:10", updated_at: "2015-09-08 02:08:10">]

```



# Take and Pluck

```
irb(main):001:0> Person.take
  Person Load (1.0ms) SELECT "people".* FROM "people" LIMIT 1
=> #<Person id: 1, first_name: "Joe", last_name: "Smith", created_at: "2015-09-08 02:08:10", updated_at: "2015-09-08 02:08:10">
irb(main):002:0> Person.take 2
  Person Load (0.2ms) SELECT "people".* FROM "people" LIMIT 2
=> [#<Person id: 1, first_name: "Joe", last_name: "Smith", created_at: "2015-09-08 02:08:10", updated_at: "2015-09-08 02:08:10">,
  #<Person id: 2, first_name: "John", last_name: "Doe", created_at: "2015-09-08 02:09:11", updated_at: "2015-09-08 02:09:11">]
irb(main):003:0> Person.all.map { |person| person.first_name }
  Person Load (0.2ms) SELECT "people".* FROM "people"
=> ["Joe", "John", "Jane"]
irb(main):004:0> Person.pluck(:first_name)
  (0.2ms) SELECT "people"."first_name" FROM "people"
=> ["Joe", "John", "Jane"]
```



# Retrieve / Read (CRUD)

✧ `where(hash)`

- Enables you to **supply conditions** for your search
- Returns `ActiveRecord::Relation` (same as `all`), but you can always **narrow** it down with `first` or treat it like an Array...

# Retrieve / Read (CRUD) - where

```
irb(main):001:0> Person.where(last_name: "Doe")
  Person Load (0.1ms) SELECT "people".* FROM "people" WHERE "people"."last_name" = ? [["last_name", "Doe"]]
=> #<ActiveRecord::Relation [#<Person id: 2, first_name: "John", last_name: "Doe", created_at: "2015-09-08 02:09:11", updated_at: "2015-09-08 02:09:11">, #<Person id: 3, first_name: "Jane", last_name: "Doe", created_at: "2015-09-08 02:11:18", updated_at: "2015-09-08 02:11:18">]>
irb(main):002:0> Person.where(last_name: "Doe").first
  Person Load (0.2ms) SELECT "people".* FROM "people" WHERE "people"."last_name" = ? ORDER BY "people"."id" ASC LIMIT 1 [["last_name", "Doe"]]
=> #<Person id: 2, first_name: "John", last_name: "Doe", created_at: "2015-09-08 02:09:11", updated_at: "2015-09-08 02:09:11">
irb(main):003:0> Person.where(last_name: "Doe")[0]
  Person Load (0.2ms) SELECT "people".* FROM "people" WHERE "people"."last_name" = ? [["last_name", "Doe"]]
=> #<Person id: 2, first_name: "John", last_name: "Doe", created_at: "2015-09-08 02:09:11", updated_at: "2015-09-08 02:09:11">
irb(main):004:0> Person.where(last_name: "Doe").pluck(:first_name)
  (0.2ms) SELECT "people"."first_name" FROM "people" WHERE "people"."last_name" = ? [["last_name", "Doe"]]
=> ["John", "Jane"]
```





# Find\_by

✧ `find_by(conditions_hash)`

- Same as `where`, but returns a single result or nil if a record with the specified conditions is not found

✧ `find_by!(conditions_hash)`

- Same as `find_by`, but throws an exception if cannot find the result

# Find\_by and Find\_by!

```
irb(main):001:0> Person.find_by(last_name: "Doe")
  Person Load (0.2ms) SELECT "people".* FROM "people" WHERE "people"."last_name" = ? LIMIT 1 [["last_name", "Doe"]]
=> #<Person id: 2, first_name: "John", last_name: "Doe", created_at: "2015-09-08 02:09:11", updated_at: "2015-09-08 02:09:11">
irb(main):002:0> Person.where(last_name: "Doe")
  Person Load (0.3ms) SELECT "people".* FROM "people" WHERE "people"."last_name" = ? [["last_name", "Doe"]]
=> #<ActiveRecord::Relation [#<Person id: 2, first_name: "John", last_name: "Doe", created_at: "2015-09-08 02:09:11", updated_at: "2015-09-08 02:09:11">, #<Person id: 3, first_name: "Jane", last_name: "Doe", created_at: "2015-09-08 02:11:18", updated_at: "2015-09-08 02:11:18">]>
irb(main):003:0> Person.find_by(last_name: "Nosuchdude")
  Person Load (0.4ms) SELECT "people".* FROM "people" WHERE "people"."last_name" = ? LIMIT 1 [["last_name", "Nosuchdude"]]
=> nil
irb(main):004:0> Person.find_by!(last_name: "Incognito")
  Person Load (0.2ms) SELECT "people".* FROM "people" WHERE "people"."last_name" = ? LIMIT 1 [["last_name", "Incognito"]]
ActiveRecord::RecordNotFound: Couldn't find Person
```



# limit / offset

✧ `limit(n)`

- Enables you to **limit** how many records come back

✧ `offset(n)`

- Don't start from the beginning; skip a few

✧ You can **combine** these two to “page” through large collections of records in your database

# limit / offset

```
irb(main):001:0> Person.count
  (0.1ms) SELECT COUNT(*) FROM "people"
=> 3
irb(main):002:0> Person.all.map { |person| "#{person.first_name} #{person.last_name}" }
  Person Load (0.3ms) SELECT "people".* FROM "people"
=> ["Joe Smithson", "John Doe", "Jane Smithie"]
irb(main):003:0> Person.offset(1).limit(1).map { |person| "#{person.first_name} #{person.last_name}" }
  Person Load (0.2ms) SELECT "people".* FROM "people" LIMIT 1 OFFSET 1
=> ["John Doe"]
irb(main):004:0> Person.offset(1).limit(1).all.map { |person| "#{person.first_name} #{person.last_name}" }
  Person Load (0.1ms) SELECT "people".* FROM "people" LIMIT 1 OFFSET 1
=> ["John Doe"]
```



# Summary

- ✧ ActiveRecord is very intuitive when it comes to interacting with a database
- ✧ Always keep in mind if you are getting back a single result (`find`) or an `ActiveRecord::Relation` (`where`)

## What's Next?

- ✧ Active Record CRUD continued

