In this lecture, we will discuss…

- ✧ Active Record **CRUD**
  - **U**pdate
  - **D**elete

Update (CR*U*D)

- Two ways to update a record in the database:
  1. Retrieve a record, modify the values and then call `save`
  2. Retrieve a record and then call `update` method passing in a hash of attributes with new values

- There is also `update_all` for batch updates
  - You can chain this to the end of `where`

Update (CR*U*D)

```
irb(main):001:0> jane = Person.find_by first_name: "Jane"
  Person Load (0.2ms)  SELECT  "people".* FROM "people" WHERE "people"."first_name" = ? LIMIT 1  [["first_name", "Jane"]]
=> #<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> jane.last_name = "Smithie"
=> "Smithie"
irb(main):003:0> jane.save
   (0.2ms)  begin transaction
  SQL (0.4ms)  UPDATE "people" SET "last_name" = ?, "updated_at" = ? WHERE "people"."id" = ?  [["last_name", "Smithie"], ["upda
ted_at", "2015-09-08 02:59:20.775235"], ["id", 3]]
   (1.4ms)  commit transaction
=> true
irb(main):004:0> jane = Person.find(3)
  Person Load (0.2ms)  SELECT  "people".* FROM "people" WHERE "people"."id" = ? LIMIT 1  [["id", 3]]
=> #<Person id: 3, first_name: "Jane", last_name: "Smithie", created_at: "2015-09-08 02:11:18", updated_at: "2015-09-08 02:59:2
0">
irb(main):005:0> Person.find_by(last_name: "Smith").update(last_name: "Smithson")
  Person Load (0.2ms)  SELECT  "people".* FROM "people" WHERE "people"."last_name" = ? LIMIT 1  [["last_name", "Smith"]]
   (0.1ms)  begin transaction
  SQL (0.3ms)  UPDATE "people" SET "last_name" = ?, "updated_at" = ? WHERE "people"."id" = ?  [["last_name", "Smithson"], ["upd
ated_at", "2015-09-08 03:00:33.037717"], ["id", 1]]
   (1.4ms)  commit transaction
=> true
```

Delete (CRU*D*)

- `destroy(id)` or `destroy`
  - Removes a particular instance from the DB
  - Instantiates an object first and performs callbacks before removing
  - See http://guides.rubyonrails.org/active_record_callbacks.html
- `delete(id)`
  - Removes the row from DB
- There is also a `delete_all` CAREFUL!

Delete / Destroy

```
irb(main):001:0> Person.count
   (0.1ms)  SELECT COUNT(*) FROM "people"
=> 3
irb(main):002:0> jane = Person.find_by first_name: "Jane"
  Person Load (0.2ms)  SELECT  "people".* FROM "people" WHERE "people"."first_name" = ? LIMIT 1  [["first_name", "Jane"]]
=> #<Person id: 3, first_name: "Jane", last_name: "Smithie", created_at: "2015-09-08 02:11:18", updated_at: "2015-09-08 02:59:20">
irb(main):003:0> jane.destroy
   (0.2ms)  begin transaction
  SQL (0.5ms)  DELETE FROM "people" WHERE "people"."id" = ?  [["id", 3]]
   (1.5ms)  commit transaction
=> #<Person id: 3, first_name: "Jane", last_name: "Smithie", created_at: "2015-09-08 02:11:18", updated_at: "2015-09-08 02:59:20">
irb(main):004:0> joe = Person.find_by first_name: "Joe"
  Person Load (0.1ms)  SELECT  "people".* FROM "people" WHERE "people"."first_name" = ? LIMIT 1  [["first_name", "Joe"]]
=> #<Person id: 1, first_name: "Joe", last_name: "Smithson", created_at: "2015-09-08 02:08:10", updated_at: "2015-09-08 03:00:33">
irb(main):005:0> Person.delete(joe.id)
  SQL (2.8ms)  DELETE FROM "people" WHERE "people"."id" = ?  [["id", 1]]
=> 1
irb(main):006:0> Person.count
   (0.2ms)  SELECT COUNT(*) FROM "people"
=> 1
```

Summary

- Update and Delete are both simple to use
- Delete has a
  - 1. “go straight to DB version” (`delete`)
  - 2. Instantiate Ruby object and let it interact with DB when it’s ready version (`destroy`)