Lecture

ORM Review / Intro to ActiveRecord

* Review ORM assignment from the previous day
* Intro to ActiveRecord as an ORM
* Query Methods
* Validations
* Associations

ACTIVE RECORDS –

first step – require active record, pry and sqlite3

then setup a database logger – arlog – this will create a new file called database.log – its logging everything you do to the project.

ActiveRecord::Base.logger

then establish a connection with active record – arconn

ActiveRecord::Base.establish\_connection(

adapter: ‘sqlite3’, -- specify the adapter, in our assignment we use postgres

database: ‘sample.db’

)

3rd step – Define a schema – arschema. 🡪 schemas are the specification into our database

ActiveRecord::Schema.define do

create\_table :shows, force: true do |t| 🡪

validates :name, presence: true, length: { minimum: 3 }

t.string :name

end 🡪 ##force: true is a special keyword says if this table exists in the database, destroy it and recreate it

create\_table :episodes, force: true do |t|

t.string :name

t.belongs\_to :show

end

end

## active record will figure out the primary key for us.

4th step – Create classes that inherit from active record base – arclass

class Show < ActiveRecord::Base 🡪 (notice that when we created the table earlier it was showS but now its show)

has\_many :episodes

end

class Episode < ActiveRecord::Base

belongs\_to :show, required: true 🡪 says there needs to be a show to add an episode

end

5th step -- Create data – ardata

bing\_bang\_theory = Show.create(name: “Big Bang”)

bing\_bang\_theory.episodes.create(name: “Fly to the moon”)

bing\_bang\_theory.episodes.create(name: “Jump higher”)

bing\_bang\_theory.episodes.create(name: “Sheldon”)

\*\*\* if you created a database and you wanna see the tables. go to terminal – sqlite3 databasename.db

.mode column then .headers on then .tables 🡪 it will show the tables then—use sql .. SELECT \* FROM shows;

etc.

then in sqlite3 we did the INSERT INTO (….) VALUES (…..)

One of the good things about active record is that we can add constraints to our columns. look at the line below create\_table in ActiveRecord::Schema.def --- these are called validations

\*\*\*\* The bang operator at the end of a method raises an error and rescue if it fails. Example:

show.save

begin

show.find(10)!

resucue

ActiveRecord::RecordNotFound

# maybe create a repl like episode doesn’t exist wanna create a new one??

end

google activerecord validations.

\*\* t.timestamp creates a created\_at and updated\_at columns \*\*