In this lecture, we will discuss…

- `where`
- `count`
- `distinct`
- `exists`
- `geolocation`
- `first_or_create_by`
- `first_or_initialize_by`

where (count, distinct)

- `Movie.where(:title => "Rocky").count`
- `Movie.where(:year.gt => 2000).distinct(:title)`

Note: Schema Design lecture reference: year should be a number – we can use ‘`gt`’ or ‘`lt`’ queries

where (first)

- Get the first document
- If no sort options are provided, Mongoid will add an ascending _id sort to the criteria
- `Movie.first`
- `Movie.where(:rated=> "R").first`

`where (exists?)`

- Determine if any documents exist in the database
- Will return true for 1 or more
- `Movie.exists?`
- `Movie.where(:title => "Titanic").exists?`

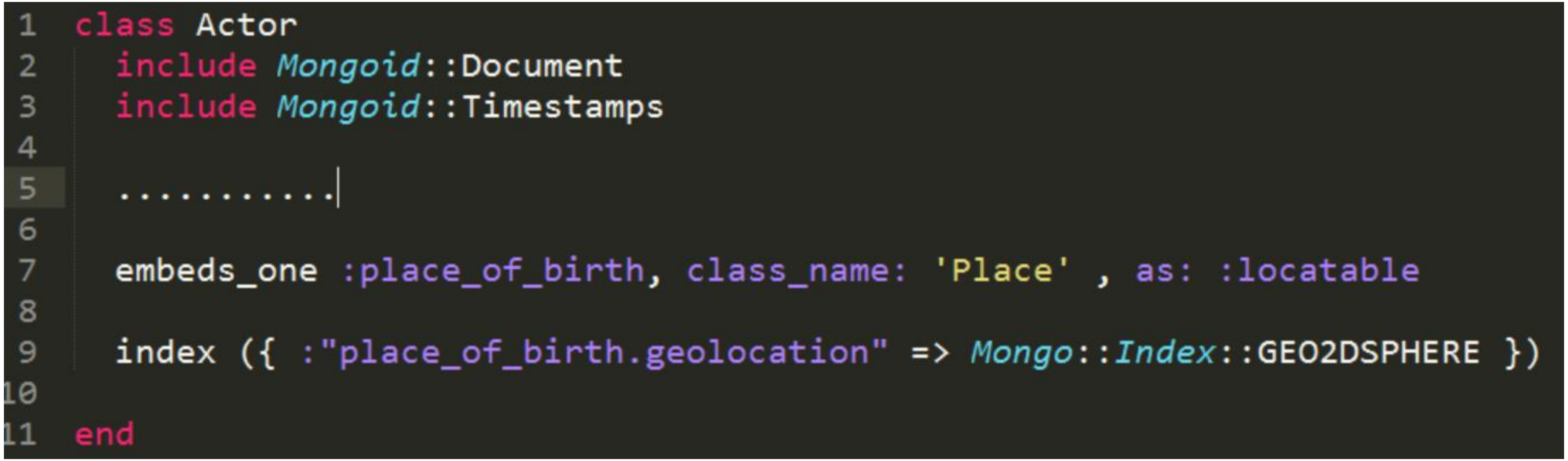
where – :$exists and :$regex

-
  ```
  writer=Writer.where(:hometown=>{
        :$exists=>0}).first
  ```

-
  ```
  damon=Actor.where(:name=>{
        :$regex=>"Matt Da"}).first
  ```

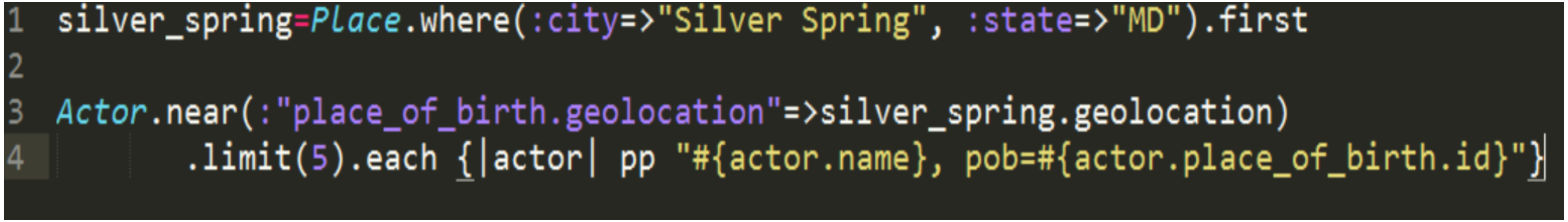
`where` – Geolocation query

- In Mongoid, every model class has the built-in ability to express and execute a Geolocation query.

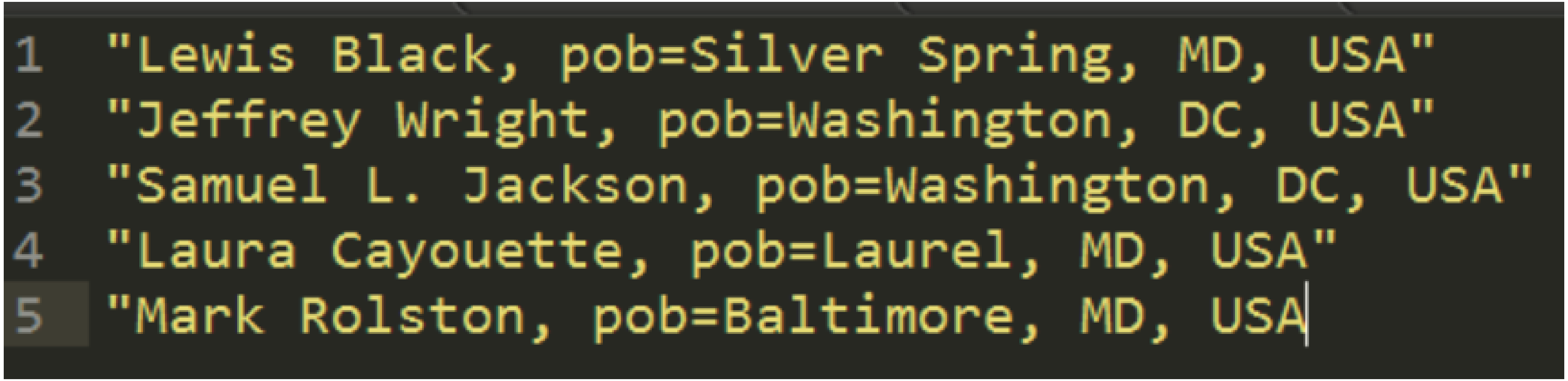


```
class Actor
  include Mongoid::Document
  include Mongoid::Timestamps

  ...........

  embeds_one :place_of_birth, class_name: 'Place' , as: :locatable

  index ({ :"place_of_birth.geolocation" => Mongo::Index::GEO2DSPHERE })

end
```

where – Geolocation query



```
silver_spring=Place.where(:city=>"Silver Spring", :state=>"MD").first

Actor.near(:"place_of_birth.geolocation"=>silver_spring.geolocation)
        .limit(5).each {|actor| pp "#{actor.name}, pob=#{actor.place_of_birth.id}"}
```



```
"Lewis Black, pob=Silver Spring, MD, USA"
"Jeffrey Wright, pob=Washington, DC, USA"
"Samuel L. Jackson, pob=Washington, DC, USA"
"Laura Cayouette, pob=Laurel, MD, USA"
"Mark Rolston, pob=Baltimore, MD, USA
```

where - first_or_create

- ✧ Find the first document by the provided attributes
- ✧ If not found, create and return a newly persisted one
- ✧ `Movie.where(:title => "Rocky20").first_or_create`

where - first_or_initialize

- Find the first document by the provided attributes
- If not found, instantiate and return a new one
- `Movie.where(:title => "Rocky21").first_or_initialize`

Summary

- `where` – with some additional criteria

What’s Next?

- `pluck` and `scope`