associations in Rails

Daniel Jackson
ActiveRecord, an “object relational mapper”
› generates class methods from database schema
› backs up object methods with SQL updates/queries

essential idea
› row database table = object in model class

<table>
<thead>
<tr>
<th>reviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

reviews table in database

<table>
<thead>
<tr>
<th>id</th>
<th>rating</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5</td>
<td>yummy!</td>
</tr>
</tbody>
</table>

object in Review class
multiple tables & joins

can make multiple tables
› can traverse from table to table
› just use `Review.find` etc

but this is clumsy!
› find all reviews by user u
  
  \[
  \text{id} = u.\text{id}  \\
  \text{reviews} = \text{Review.where(by: id)}
  \]

a better way
› traverse associations
  
  \[
  \text{reviews} = u.\text{reviews}
  \]

---

show content and ratings of reviews by user with uid

```
select content, rating from reviews 
where reviews.by = uid
```
declaring associations

```ruby
class User < ActiveRecord::Base
  has_many :reviews
end

class Review < ActiveRecord::Base
  belongs_to :user
end
```

with these declarations, can now use expressions like

- `u.reviews`: an array of review objects
- `r.user`: a single user object
what does belongs_to declaration do here?
› creates a foreign key called user in the reviews table
› this corresponds to what we called “by” before
has_one vs belongs_to

```ruby
class User < ActiveRecord::Base
  has_one :account
end

class Account < ActiveRecord::Base
  belongs_to :user
end
```

when association is 1:1
• put belongs_to on one end
• put has_one on the other end

what's the difference?
• belongs_to appears in migration
  and creates the foreign key
many-to-many associations

```ruby
class Student < ActiveRecord::Base
  has_and_belongs_to_many :courses
end

class Course < ActiveRecord::Base
  has_and_belongs_to_many :students
end
```

in a registrar’s system
› Student and Course models
› m:m association between them
› want to write `student.courses, course.students`
in the migration

class CreateStudentsAndCourses < ActiveRecord::Migration
  def change
    create_table :students do |t|
      t.string :name
      t.timestamps
    end

    create_table :courses do |t|
      t.string :number
      t.timestamps
    end

    create_table :students_courses do |t|
      t.belongs_to :student
      t.belongs_to :course
    end
  end
end

need to create an association table
› often better to surface as a model class itself
› eg, Registration
other features of associations

propagating deletions
› eg, can delete all reviews when authoring user deleted through associations
› eg, connect subjects to users via reviews

polymorphic associations
› object can belong to different model classes
› eg, picture belongs to employee or to product

self joins
› model can belong to itself
› eg, employee has manager and subordinates associations