Rails Spring 2014

week 4

Homework Review

- Review the parts of Class Management we did last week that the homework coverd

Agenda

- Active Record
  - Migrations - review adding email
  - Relationships
  - Active Record Query Interface
  - Scopes
- Databases

Active Record

- Rails ORM - Object Relational Mapping
  - How we get data into Ruby objects
  - Translates between the rows in the table and Ruby objects

Migrations

- Ruby DSL for SQL translation
- Add, Remove, Update DB tables/columns
- Can put in DB constraints, defaults, and indexes - but make sure your app reflects them!
- Always forward thinking - once it's been run in multiple environments you can easily mess-up your app if you edit a migration

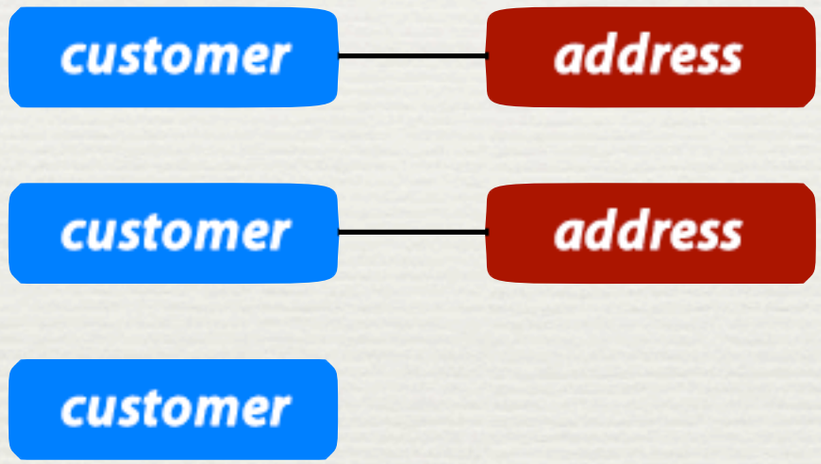
Adding Courses

- We want students to be able to manage their course
- Client tells us that each student has a list of courses they are taking, they can enter them and manage (CRUD) them

Relationships

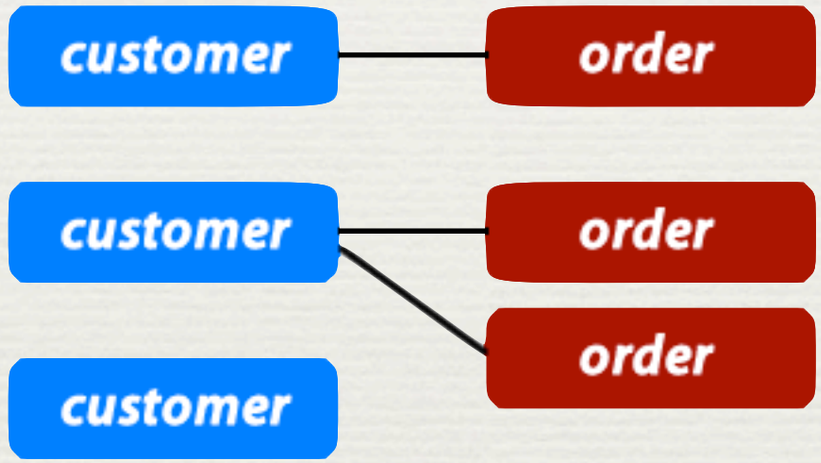
- 3 Types of Relationships
  - one-to-one
  - one-to-many
  - many-to-many

One-To-One



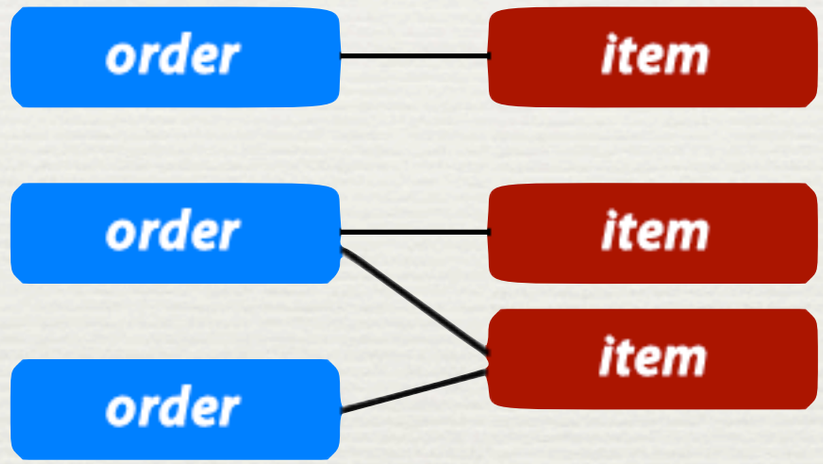
http://code.tutsplus.com/articles/sql-for-beginners-part-3-database-relationships--net-8561

One-To-Many



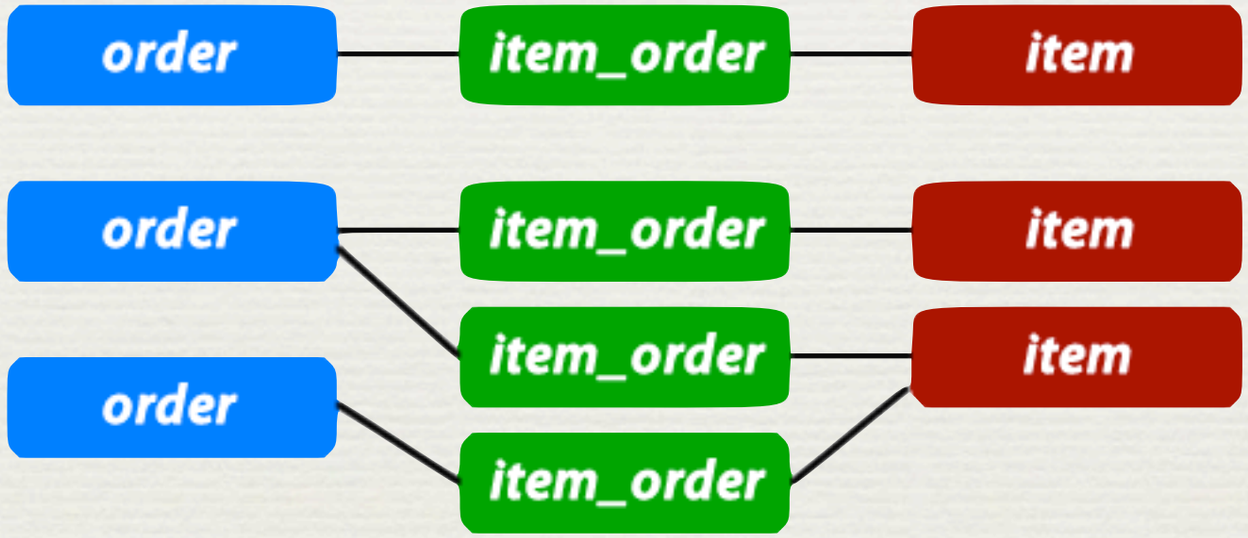
http://code.tutsplus.com/articles/sql-for-beginners-part-3-database-relationships--net-8561

Many-To-Many



http://code.tutsplus.com/articles/sql-for-beginners-part-3-database-relationships--net-8561

Many-To-Many



http://code.tutsplus.com/articles/sql-for-beginners-part-3-database-relationships--net-8561

One-To-Many

- Students have many Courses

Add Courses

- rails g model Course title description:text student_id:integer

Add Relationship

- Student
  - has_many :courses
- Course
  - belongs_to :course

Rails Console Demo

- s = Student.first
- s.courses
- s.courses.create title: ‘Rails is cool!’

Requirements Change!

- We have a bunch of duplicate courses!
- Turns out Courses have many Students as well!
- What to do!?

Change the Relationship

- Easy!
- 2 Ways of Many-to-Many in Rails, Only ONE is Correct!! :)

habtm - Evil, Bad, Ugly

- don't use has and belongs to many
- Why?

I Forgot!

- Every Student has a unique registration id for each course they are in
- Where ya gonna put that with your silly habtm??!!! ;)

Rails Many-to-Many

- Create Registration or StudentCourses model
  - rails g model Registration student_id:integer course_id:integer number
- Student
  - has_many :registrations
  - has_many :courses, through: : registrations

Rails Console Demo

- s = Student.first
- s.courses
- s.courses.create title: ‘Rails is Super Magic!’

AR QI

- Active Record Query Interface!
  - http://guides.rubyonrails.org/active_record_querying.html
- How to do SQL in Rails

Named Scopes

- Save AR query for a model
- Student.renee
- Student.teachers.names
- s = Student.first
- s.courses.rails

Named Scopes

- Student
  - scope :renee, -> { where full_name: 'Renee' }
  - scope :teachers, -> { where profession: 'Teacher' }
  - scope :names, -> { pluck :full_name }
- Courses
  - scope :rails, -> { where('title LIKE ?', '%Rails%') }

Databases

- Gemfile DB adaptor (can specify different dbs for different groups/environments)
- config/databases.yml - specify adaptor and credentials
- rake tasks help you setup, migrate, create, and drop
- Switching DBs is trivial!
- JRuby on Rails supports JDBC

Seed file

- Setup data
  - ex: first user, drop down lists, known courses
- rake db:seed

Homework

- Reading:
  - Chapter 6: Sections 4-6
  - Chapter 7: All
  - Chapter 9: Section 1
  - Chapter 11: Sections 9-11
- Code:
  - Add 2 relationships to your app:
    - 1. Has Many (one-to-many)
    - 1. Has Many Through (many-to-many)
  - Add 2 named scopes