

# UW Ruby Programming 110

## Winter 2015

## Michael Cohen

## Lecture 9

## Mar 5, 2015

# Lecture 9

**1. ActiveRecord**

**2. Production Ruby**

# Section 1

# ActiveRecord

## Section 1: ActiveRecord

### What is it?

**ActiveRecord is the Ruby-on-Rails ORM**

**Object  
Relational  
Mapper**

## Section 1: ActiveRecord

### Purpose

- Represent models and their data
- Represent associations between these models
- Validate models before they get persisted
- Perform db operations in an object-oriented fashion
- Represent inheritance hierarchies through related models

# Section 1: ActiveRecord

## Overview

- 1. Naming Conventions**
- 2. Schema Conventions**
- 3. Creating Models**
- 4. CRUD**
- 5. Validations**
- 6. Migrations**
- 7. Relationships**

# Section 1: ActiveRecord

## Naming Conventions

### Database Table

- Plural with underscores separating words (e.g., `book_clubs`).

### Model Class

- Singular with the first letter of each word capitalized (e.g., `BookClub`).

## Section 1: ActiveRecord

# Schema Conventions: Primary keys

**By default, ActiveRecord will use an integer column named `id` as the table's primary key.**



## Section 1: ActiveRecord

# Schema Conventions: Foreign keys

**These fields should be named following the pattern `singularized_table_name_id` (e.g., `item_id`, `order_id`).**

**These are the fields that Active Record will look for when you create associations between your models.**

## Section 1: ActiveRecord

### Schema Conventions: datetime columns

`created_at`

**Automatically gets set to the current date and time when the record is first created.**

`updated_at`

**Automatically gets set to the current date and time whenever the record is updated.**

## Section 1: ActiveRecord

### Schema Conventions: date columns

`created_on`

**Automatically gets set to the current date when the record is first created.**

`updated_on`

**Automatically gets set to the current date whenever the record is updated.**

## Section 1: ActiveRecord

# Creating Models: ruby class

```
class Product < ActiveRecord::Base  
  # ...  
end
```

## Section 1: ActiveRecord

### Creating Models: db table

```
CREATE TABLE products (  
  id int(11) NOT NULL auto_increment,  
  name varchar(255),  
  PRIMARY KEY (id)  
);
```

## Section 1: ActiveRecord

### CRUD: Create

```
user = User.create(  
    name: "David",  
    occupation: "Code Artist"  
)
```

## Section 1: ActiveRecord

### CRUD: Create

```
user = User.new  
user.name = "David"  
user.occupation = "Code Artist"  
user.save
```

## Section 1: ActiveRecord

### CRUD: Read

```
# return a collection with all users:
```

```
users = User.all
```

```
# return the first user:
```

```
user = User.first
```



## Section 1: ActiveRecord

### CRUD: Read

```
# return the first user named David  
david = User.find_by(name: 'David')
```

## Section 1: ActiveRecord

### CRUD: Read

```
# find all users named David
# who are Code Artists
# sort by created_at in reverse chronological order
users = User.where(
    name: 'David',
    occupation: 'Code Artist')
.order('created_at DESC')
```

## Section 1: ActiveRecord

### CRUD: Update

```
user = User.find_by(name: 'David')  
user.name = 'Dave'  
user.save
```

## Section 1: ActiveRecord

### CRUD: Update

```
user = User.find_by(name: 'David')  
user.update(name: 'Dave')
```

## Section 1: ActiveRecord

# CRUD: Update

```
user = User.find_by(name: 'David').update_all(name: 'Dave')
```

## Section 1: ActiveRecord

### CRUD: Delete

```
user = User.find_by(name: 'David')  
user.destroy
```

# Section 1: ActiveRecord

## Validations

```
class User < ActiveRecord::Base
  validates :name, presence: true
end
```

```
user = User.new
user.save # => false
user.save! # => ActiveRecord::RecordInvalid: Validation failed: Name can't be blank
```

# Section 1: ActiveRecord

## Migrations

```
class CreatePublications < ActiveRecord::Migration
  def change
    create_table :publications do |t|
      t.string :title
      t.text :description
      t.integer :publisher_id
      t.boolean :single_issue

      t.timestamps null: false
    end
    add_index :publications, :title
  end
end
```



# Section 1: ActiveRecord Relationships

```
class Company < ActiveRecord::Base
  has_many :employees
  # ...
end
```

```
class Employee < ActiveRecord::Base
  belongs_to :company
  # ...
end
```

## Section 1: ActiveRecord Relationships

```
company = Company.first  
company.employees.count
```

```
employee1 = company.employees.first  
employee1.company # => company
```

## Section 2

# Production Ruby