Rails from the Ground Up!

with Jared Richardson

RoleModel Software

Rails
Database Migrations

Did you Create a Template?

```
rails generate model card_deck
```

What Happened?

```
      invoke  active_record

      create    db/migrate/
20120213203148_create_card_decks.rb

      create    app/models/card_deck.rb

      invoke    test_unit

      create      test/unit/card_deck_test.rb

      create      test/fixtures/card_decks.yml
```

Database Migrations

Look in war/db/migrate/

```
class CreateCardDecks < ActiveRecord::Migration
  def change
    create_table :card_decks do |t|
      t.timestamps
    end
  end
end
```

Migrations

Make incremental database change

Formerly had "up" and "down"

To Add and Drop

Now "change" does both

You can use either

Common Operations

add_column

add_index

change_column

change_table

create_table

drop_table

remove_column

remove_index

rename_column

For Instance

```
def change
  create_table :products do |t|
    t.string :name
    t.text :description

    t.timestamps
  end
end
```

Your Turn

Add a name attribute to your CardDeck

Edit the migration directly

NOTE: Don't do this to an applied migration!

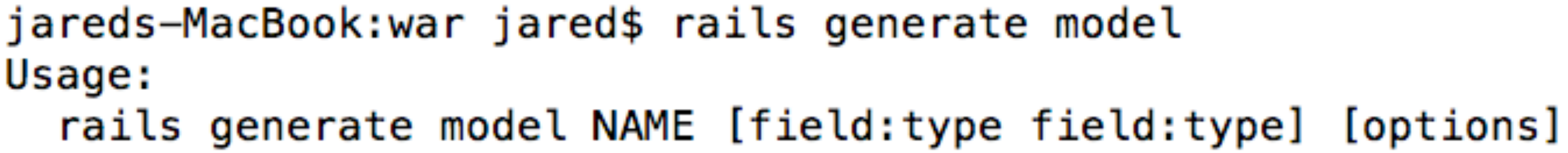
rake db:migrate

```
rake db:migrate VERSION=0

# edit the migration

rake db:migrate
```

Another Way



```
jareds-MacBook:war jared$ rails generate model
Usage:
  rails generate model NAME [field:type field:type] [options]
```

Remove the Model

```
rails destroy model Card_Deck
```

Migrations for Free!

```
rails generate model

Card_Deck name:string

OR

Card_Deck name:string hands_played:integer
```

Stand Alone Migrations

```
rails generate migration AddCardCountToCardDeck
```

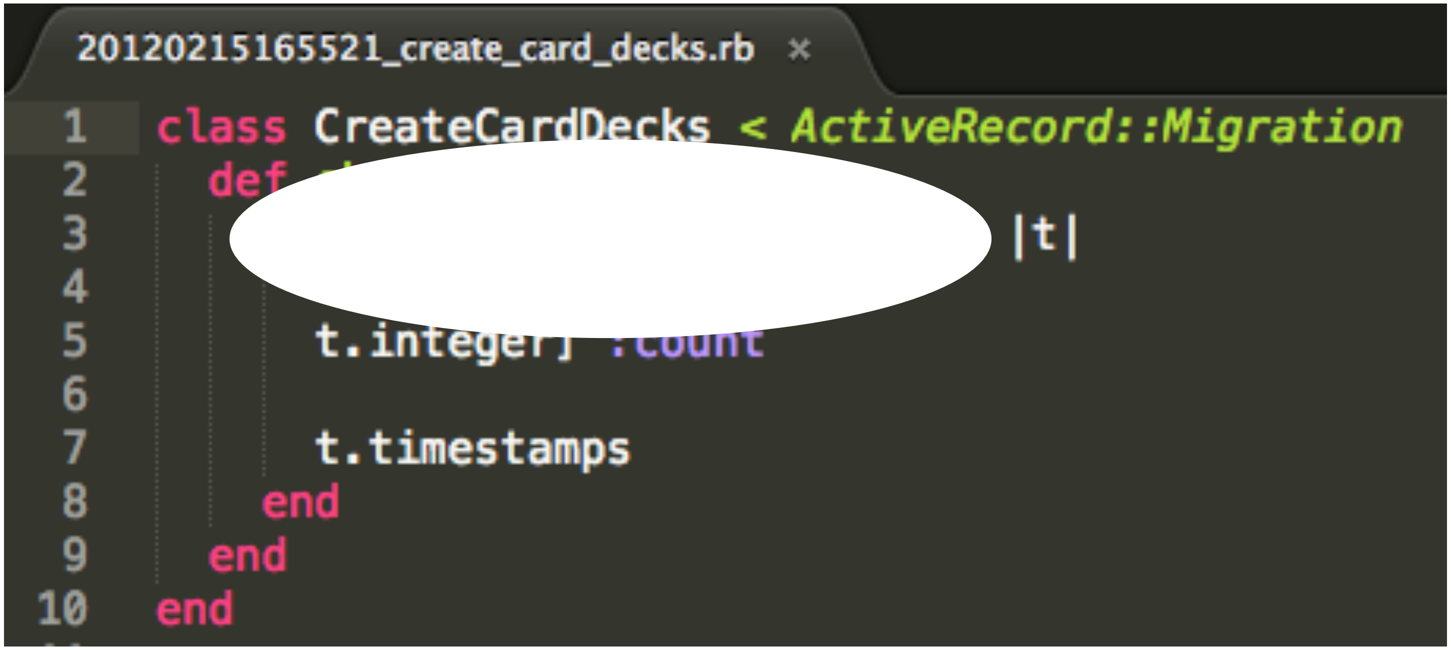
Run this

What did you get?

Better Stand Alone Migrations

```
rails generate migration

AddCardCountToCardDeck card_count:integer
```

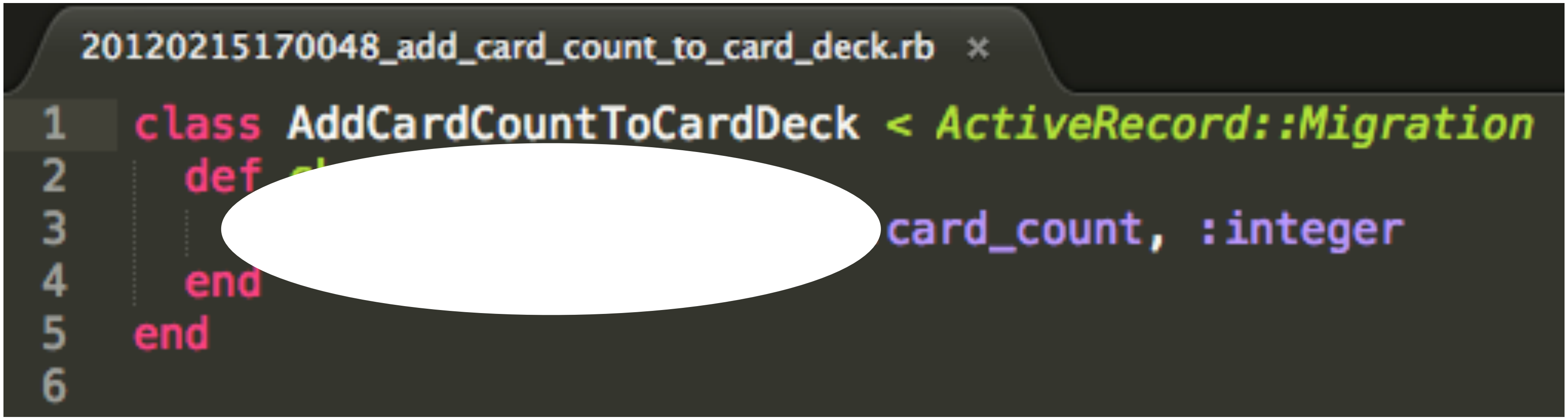


20120215165521_create_card_decks.rb

```
class CreateCardDecks < ActiveRecord::Migration
  def [redacted] |t|

      t.integer :count

      t.timestamps
    end
  end
end
```

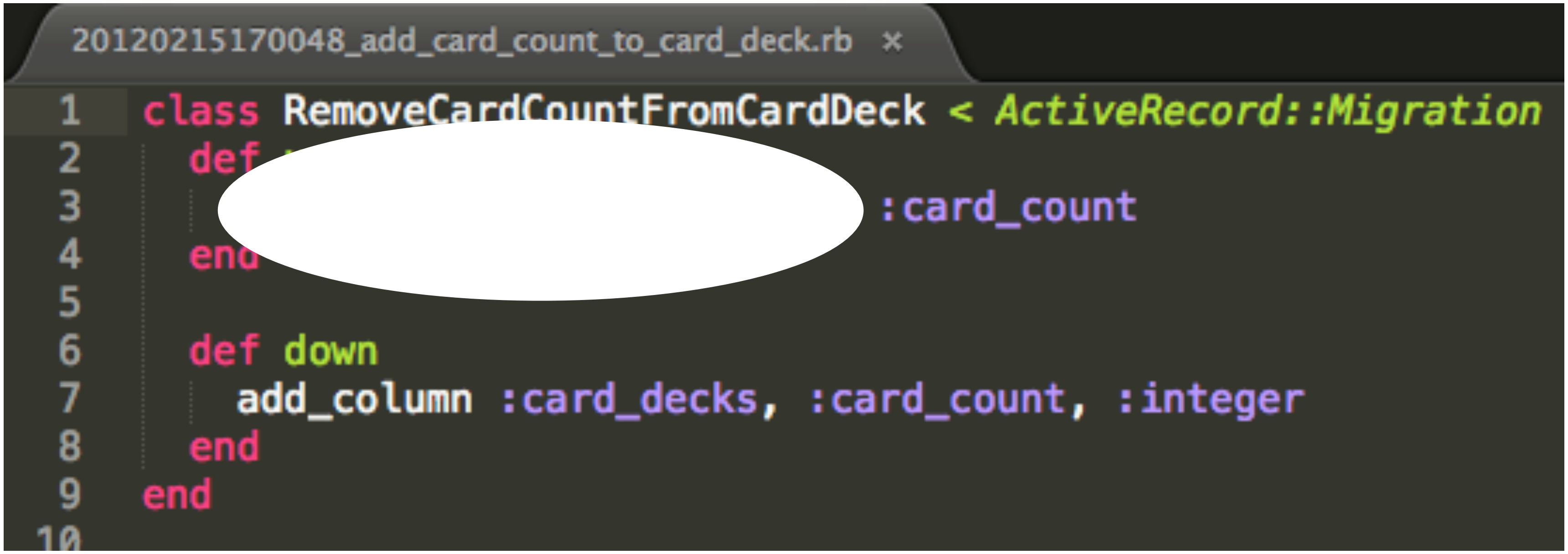


20120215170048_add_card_count_to_card_deck.rb

```
class AddCardCountToCardDeck < ActiveRecord::Migration
  def [redacted] card_count, :integer
  end
end
```

Or Removing...

```
rails generate migration
RemoveCardCountFromCardDeck card_count:integer
```



20120215170048_add_card_count_to_card_deck.rb ×

```ruby
class RemoveCardCountFromCardDeck < ActiveRecord::Migration
  def [redacted]
                                       :card_count
  end

  def down
    add_column :card_decks, :card_count, :integer
  end
end
```

Migration Smarts

```
Add<column>To<table> <name>:<type>

Add<label>To<table> <n>:<t> <n>:<t> <n>:<t>

AddDetailsToCardDeck a:string b:integer
```

change_table

```
change_table :card_deck do |t|
  t.remove :description, :name
  t.string :first_name
  t.index :part_number
  t.rename :list_name, :last_name
end
```

Magic Colums

"id" is always added

"t.timestamps" adds...

created_at & updated_at

t.references :card (card = model name)

adds "card_id"

Migrations w/SQL

```
def add_fk
  #add a foreign key
  execute <<-SQL
    ALTER TABLE cards
      ADD CONSTRAINT fk_cards_categories
      FOREIGN KEY (category_id)
      REFERENCES categories(id)
  SQL
end
```

Final Thoughts

```
rake migrate db:rollback

rake migrate db:rollback STEP=3

rake migrate db:redo STEP=4

rake db:reset

db/schema.rb
```

Next...

???