

# In this lecture, we will discuss...

- ✧ How to create and drop tables
- ✧ How to modify and rename columns



# Creating Tables with Migrations

- ✧ By convention, table names in Rails are **always** named **plural** (many rows...)
- ✧ An `id` column is **automatically created** to be used as **primary key**
- ✧ `timestamps` method creates `created_at` and `updated_at` columns
- ✧ `create_table` and `drop_table` create and drop the table



# Creating Cars Table

```
class CreateCars < ActiveRecord::Migration
  def change
    create_table :cars do |t|
      t.string :make
      t.string :color
      t.integer :year

      t.timestamps null: false
    end
  end
end
```

After rake db:migrate



```
sqlite> .schema cars
CREATE TABLE "cars" ("id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, "make" varchar, "color"
  varchar, "year" integer, "created_at" datetime NOT NULL, "updated_at" datetime NOT NULL);
sqlite> █
```

# rake db:rollback

- ✧ Undoes the **last** migration (possibly applies the **down** method)

```
~/fancy_cars$ rake db:rollback  
== 20150907153643 CreateCars: reverting =====  
-- drop_table(:cars)  
   -> 0.0011s  
== 20150907153643 CreateCars: reverted (0.0055s)
```

Oh no!!! Where did my car data go? What data?...

# Adding / Removing Column

- ✧ `add_column :table_name,`  
`:column_name, :column_type`
- ✧ `remove_column :table_name, :column_name`

Let's add a price column to the  
cars table

# Adding Price To Cars Table

```
~/fancy_cars$ rails g migration add_price_to_cars 'price:decimal{10,2}'  
  invoke  active_record  
  create  db/migrate/20150907183118_add_price_to_cars.rb
```

## FOLDERS

### ▼ fancy\_cars

#### ▶ app

#### ▶ bin

#### ▶ config

### ▼ db

#### ▼ migrate

20150907153643\_create\_cars.rb

20150907183118\_add\_price\_to\_cars.rb

20150907183118\_add\_price\_to\_cars.rb ✕

```
class AddPriceToCars < ActiveRecord::Migration  
  def change  
    add_column :cars, :price, :decimal, precision: 10, scale: 2  
  end  
end
```

Automatically detects table name  
from how we named the migration



# Adding Price To Cars Table

```
~/fancy_cars$ rake db:migrate
== 20150907153643 CreateCars: migrating =====
-- create_table(:cars)
   -> 0.0010s
== 20150907153643 CreateCars: migrated (0.0011s) =====

== 20150907183118 AddPriceToCars: migrating =====
-- add_column(:cars, :price, :decimal, {:precision=>10, :scale=>2})
   -> 0.0005s
== 20150907183118 AddPriceToCars: migrated (0.0006s) =====

~/fancy_cars$ rails db
SQLite version 3.8.5 2014-08-15 22:37:57
Enter ".help" for usage hints.
sqlite> .schema cars
CREATE TABLE "cars" ("id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, "make" varchar, "color" varchar, "
year" integer, "created_at" datetime NOT NULL, "updated_at" datetime NOT NULL, "price" decimal(10,2));
```



# Overall Schema Of Your Database

**FOLDERS**

- ▼ fancy\_cars
  - ▶ app
  - ▶ bin
  - ▶ config
- ▼ db
  - ▼ migrate
    - 20150907153643\_create\_cars.rb
    - 20150907183118\_add\_price\_to\_cars.rb
  - development.sqlite
  - schema.rb**
  - seeds.rb
- lib
- log
- public
- test
- tmp
- vendor
  - .gitignore
  - config.ru
  - Gemfile

*schema.rb*

```
# encoding: UTF-8
# This file is auto-generated from the current state of the database. Instead
# of editing this file, please use the migrations feature of Active Record to
# incrementally modify your database, and then regenerate this schema definition.
#
# Note that this schema.rb definition is the authoritative source for your
# database schema. If you need to create the application database on another
# system, you should be using db:schema:load, not running all the migrations
# from scratch. The latter is a flawed and unsustainable approach (the more migrations
# you'll amass, the slower it'll run and the greater likelihood for issues).
#
# It's strongly recommended that you check this file into your version control system.

ActiveRecord::Schema.define(version: 20150907183118) do

  create_table "cars", force: :cascade do |t|
    t.string "make"
    t.string "color"
    t.integer "year"
    t.datetime "created_at", null: false
    t.datetime "updated_at", null: false
    t.decimal "price", precision: 10, scale: 2
  end
end
```





# Renaming Columns

✧ `rename_column :table_name,  
:old_column_name, :new_column_name`

```
~/fancy_cars$ rails g migration rename_make_to_company  
  invoke  active_record  
  create  db/migrate/20150907185017_rename_make_to_company.rb
```

**FOLDERS**

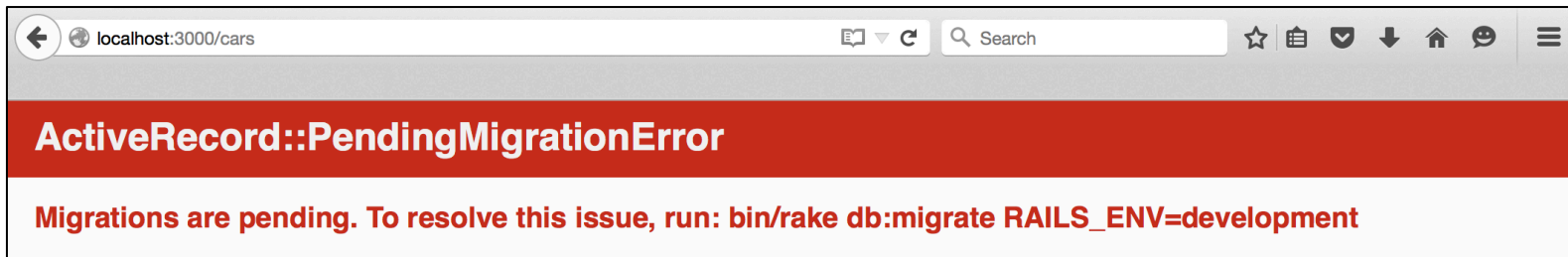
- ▼ fancy\_cars
  - ▶ app
  - ▶ bin
  - ▶ config
  - ▼ db
    - ▼ migrate
      - 20150907153643\_create\_cars.rb
      - 20150907183118\_add\_price\_to\_cars.rb
      - 20150907185017\_rename\_make\_to\_company.rb

20150907185017\_rename\_make\_to\_company.rb ✕

```
class RenameMakeToCompany < ActiveRecord::Migration
  def change
    rename_column :cars, :make, :company
  end
end
```

Added code manually






# Renaming Columns



```
~/fancy_cars$ rake db:migrate
== 20150907185017 RenameMakeToCompany: migrating ==
-- rename_column(:cars, :make, :company)
   -> 0.0036s
== 20150907185017 RenameMakeToCompany: migrated (0.0037s)
```

Does not destroy data – just renames the column...

# Renaming Columns

  localhost:3000/cars/new     Search

## NoMethodError in Cars#new

Showing /Users/kalmanhazins/fancy\_cars/app/views/cars/\_form.html.erb where line #16 raised:

undefined method `make' for #<Car:0x007f923e833f88>

Extracted source (around line #16):

```
14     <div class="field">
15       <%= f.label :make %><br>
16       <%= f.text_field :make %>
17     </div>
18     <div class="field">
19       <%= f.label :color %><br>
```

# Irreversible Migrations And More

- ✧ Sometimes, you only want your **migrations** to go **one way** (for example, when loss of data is unacceptable)
- ✧ Or, you run across any other issues related to migrations not discussed thus far
- ✧ Hit the **rails guides** before hitting Stack Overflow

<http://guides.rubyonrails.org/migrations.html>



# Summary

- ✧ Use Rails generators to **generate** migrations
- ✧ Beware of **side effects** to your app due to migrations

## What's Next?

- ✧ Dynamic Dispatch

