Module 1:

How rails interacts with the data base.

Scaffolding: 脚手架

scaffolding is this concept of being able to quickly generate code to view, delete,

and update resources

interactions happened in database

rails use SQlite

Migrations

Ruby code that lets you modify the database

Create labels,change columns etc.

The other thing that is nice about migrations迁移

is that you could create migrations, apply them to the database, which takes

the database to a different state and then you could roll back migrations.

Which means that now they go back to a previous state.

Schema

Active record

Interaction with the database data

Active record crud

Create retrieve update delete

Metaprogramming

Dynamic method

Ghost methods

Dynamic dispatch

Dynamic language

Enable developer to make decisions at runtime versus compile time

Metaprogramming

Code that generates more code or accounts for missing code

Scaffolding generator

Rails g scaffold car make color year:integer

Car entities

Columns

Make color and year

Year is an integer type

If not string type by default

Applying scaffolding to DB

Rake db:migrate

Rails s

Localhost:3000/cars

Summary

**What’s Next?**

2 Scaffolding can get you up and running quickly

2 Generates JSON response

2 Database Setup

Rails db

.help

.table

Cars schema\_migrations //show database tables as columns

.headers on

.mode columns

Select \* from cars

.exit

Migrations

Summary

**What’s Next?**

2 Introduction to Migrations

2 Rails uses SQLite by default

2 Built-in command-line DB viewer

And that version column maintains the version of the last

migration that was applied To your code and

once the migration's applied the version is stored in that scheme\_migrations table.

So, what this means is that if you run rake db:migrate more than once,

nothing's going to happen the second time because your database inside

the schema\_migrations table already has the version column and

is going to show that the latest migration has already been applied.

So, nothing is going to happen to your database.

So, what actually goes inside the ActiveRecord::Migration subclass

Anatomy of Migration

2 So, what actually goes inside the

ActiveRecord::Migration subclass?

2 Either

• def up // create a table

§ Generate db schema changes

• def down

§ Undo the changes introduced by the up method

2 Or, just change method when Rails can guess how to

undo changes (most of the time)

rake db rollback

Lecture 4

Creating Tables with Migrations

2 By convention, table names in Rails are always named

plural (many rows…)

2 An id column is automatically created to be used as

primary key

2 timestamps method creates created\_at and

updated\_at columns

2 create\_table and drop\_table create and drop the

Table

Renaming Columns

rename\_column :table\_name,

:old\_column\_name, :new\_column\_name