BACK END WEB DEVELOPMENT

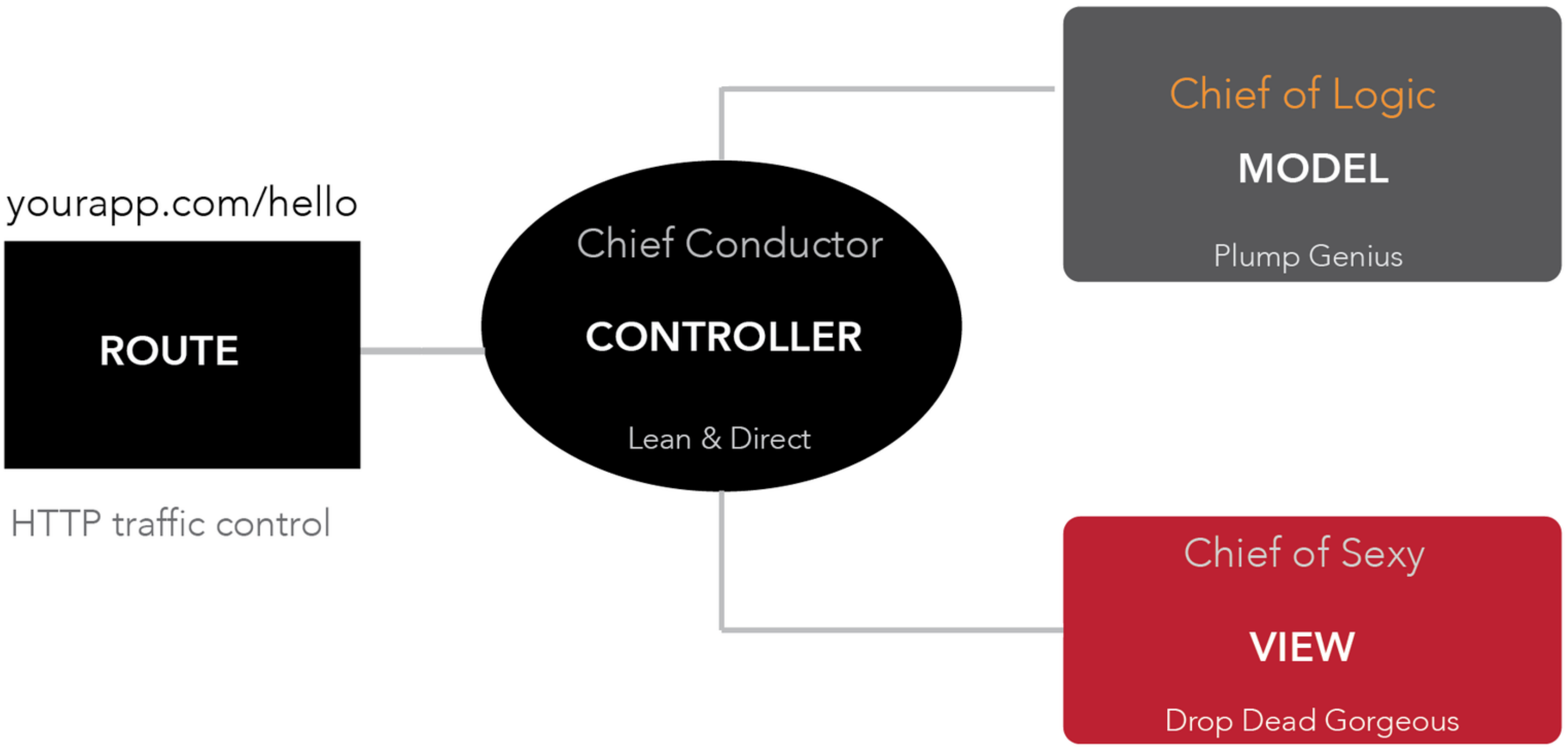
CLASS 11- SCHEMA DESIGN & RAILS ASSOCIATIONS

AGENDA

- CURRICULUM REVIEW
- CODE ALONG! MOVIE APP
- LEARN: SCHEMA DESIGN & ASSOCIATIONS
- PAIR UP! FOR SCHEMA DESIGN
- HOME WORK: THE GAME PLAN
- LAB TIME! KICK START MUSIC APP DEV

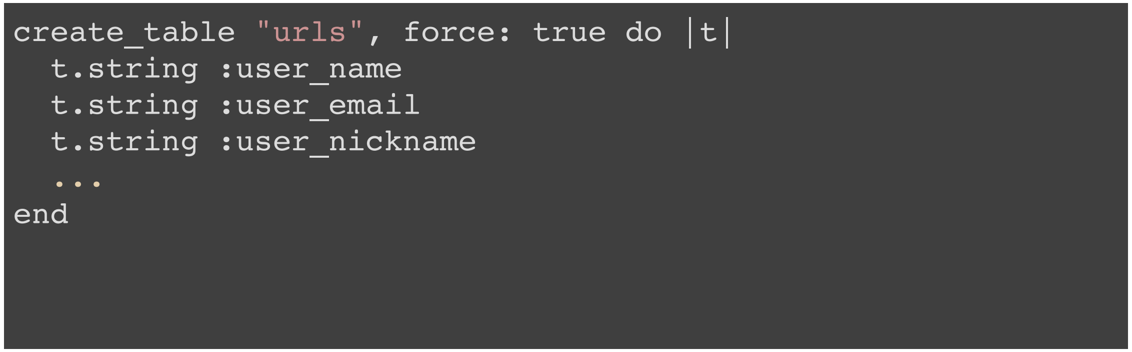
ASSOCIATIONS

ACTIVE RECORD: MODEL


yourapp.com/hello
ROUTE
HTTP traffic control
Chief Conductor
CONTROLLER
Lean & Direct
Chief of Logic
MODEL
Plump Genius
Chief of Sexy
VIEW
Drop Dead Gorgeous

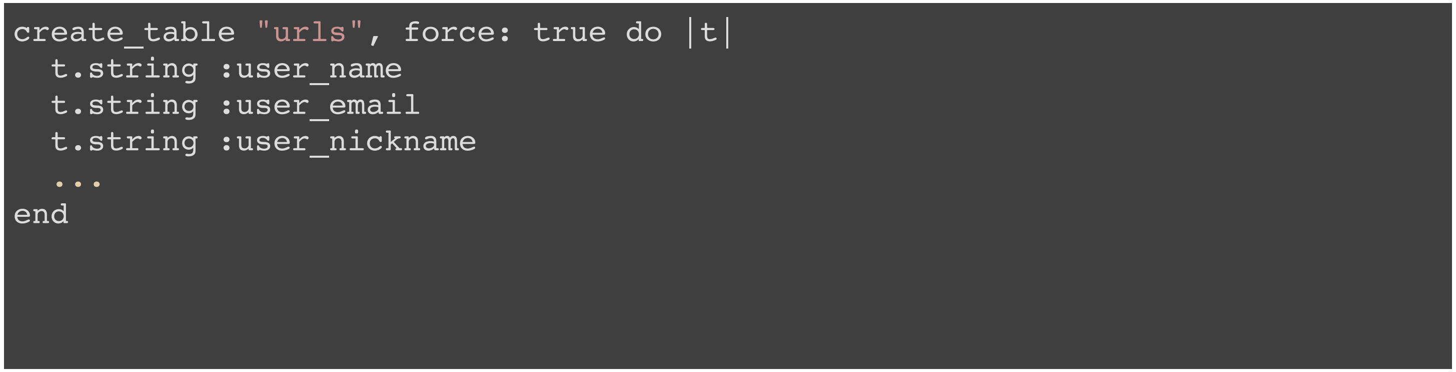
ASSOCIATIONS

- When we make Resources, sometimes they need to interact.
- For example, if we want to store the user who wrote a story, we could start by storing the user_name as an attribute on a story.
- Then we'd add user_email, user_nickname, etc… and it would quickly get messy



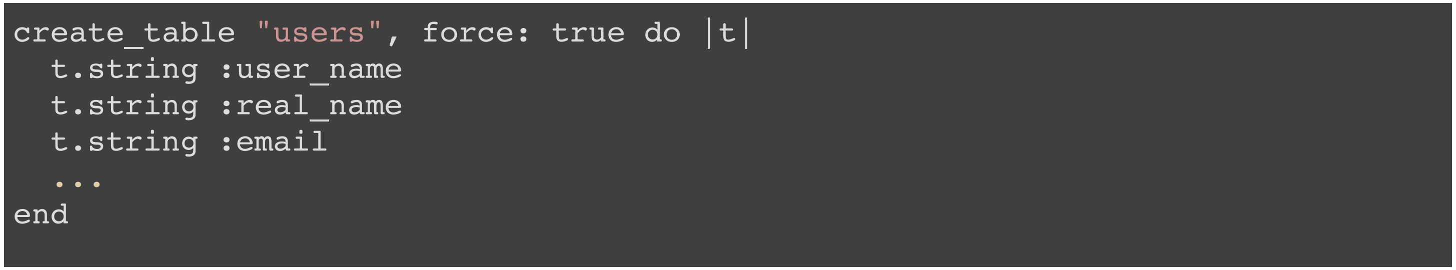
```
create_table "urls", force: true do |t|
  t.string :user_name
  t.string :user_email
  t.string :user_nickname
  ...
end
```

ASSOCIATIONS



```
create_table "urls", force: true do |t|
  t.string :user_name
  t.string :user_email
  t.string :user_nickname
  ...
end
```

This data would be better in a User table.

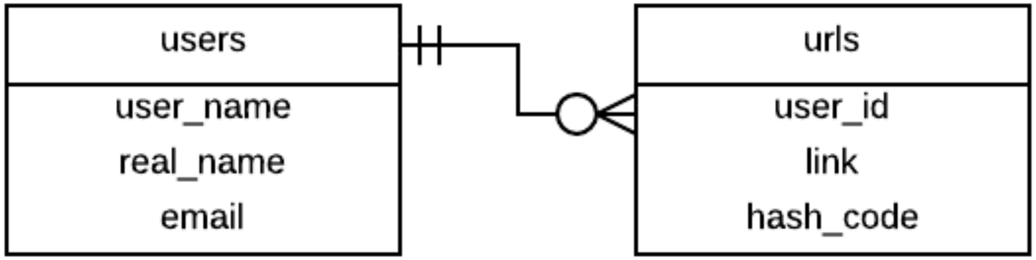


```
create_table "users", force: true do |t|
  t.string :user_name
  t.string :real_name
  t.string :email
  ...
end
```

We connect urls and users using an Association

ASSOCIATIONS: CONCEPT AND CODE

An association using an entity-relationship diagram



users

user_name

real_name

email

urls

user_id

link

hash_code

```
create_table "urls", force: true do |t|
  t.text     "link"
  t.string   "hash_code"
  t.integer  "user_id"
end

class Url < ActiveRecord::Base
  belongs_to :user
end

class User < ActiveRecord::Base
  has_many :urls
end
```

ASSOCIATIONS: KEYS

- *The **primary key** is the field that uniquely identifies each row of a table.*
- *The **foreign key** identifies which row of another table it is associate with*

```
class User < ActiveRecord::Base
  has_many :urls
end

class Url < ActiveRecord::Base
  belongs_to :user
end

user.id         # user primary key
url.id          # url primary key

url.user_id     # user foreign key
```

ASSOCIATIONS: SIX TYPES

- **belongs_to**
- **has_many**
- **has_one**
- has_and_belongs_to_many
- has_many :through
- has_one :through

QUESTIONS?

EXAMPLE!

TWITTER: SCHEMA DESIGN

SCHEMA DESIGN: TWITTER EXAMPLE

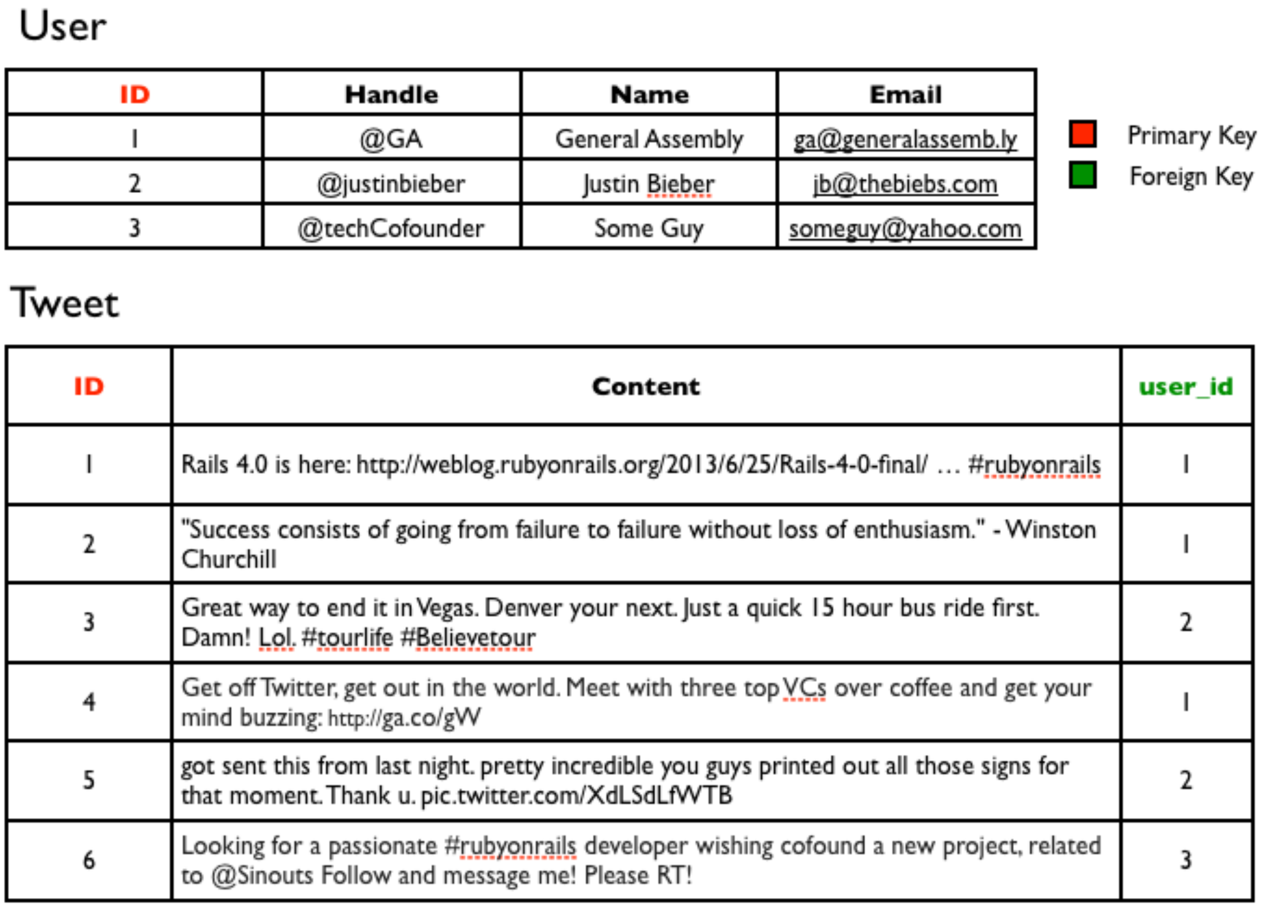
RESOURCES

- User
- Tweet
- HashTags

ASSOCIATIONS

- User has many tweets
- Tweets have many hashtags
- HashTags have many tweets

SCHEMA DESIGN: TWITTER EXAMPLE



User

ID	Handle	Name	Email
1	@GA	General Assembly	ga@generalassemb.ly
2	@justinbieber	Justin Bieber	jb@thebiebs.com
3	@techCofounder	Some Guy	someguy@yahoo.com

🟥 Primary Key

🟩 Foreign Key

Tweet

ID	Content	user_id
1	Rails 4.0 is here: http://weblog.rubyonrails.org/2013/6/25/Rails-4-0-final/ … #rubyonrails	1
2	"Success consists of going from failure to failure without loss of enthusiasm." - Winston Churchill	1
3	Great way to end it in Vegas. Denver your next. Just a quick 15 hour bus ride first. Damn! Lol. #tourlife #Believetour	2
4	Get off Twitter, get out in the world. Meet with three top VCs over coffee and get your mind buzzing: http://ga.co/gW	1
5	got sent this from last night. pretty incredible you guys printed out all those signs for that moment. Thank u. pic.twitter.com/XdLSdLfWTB	2
6	Looking for a passionate #rubyonrails developer wishing cofound a new project, related to @Sinouts Follow and message me! Please RT!	3

SCHEMA DESIGN: TWITTER EXAMPLE

```
#Migration file
create_table "tweets", force: true do |t|
  t.text     "message"
  t.integer  "user_id"
end

#Model file(s)

class User < ActiveRecord::Base
  has_many :tweets
end

class Tweet < ActiveRecord::Base
  belongs_to :user
end


#Rails Console
  user.tweets        # list all tweets associated with user
  tweet.user_id     # user foreign key with tweet table connects user to tweet
```

JOIN TABLE

Join tables connect two other tables.

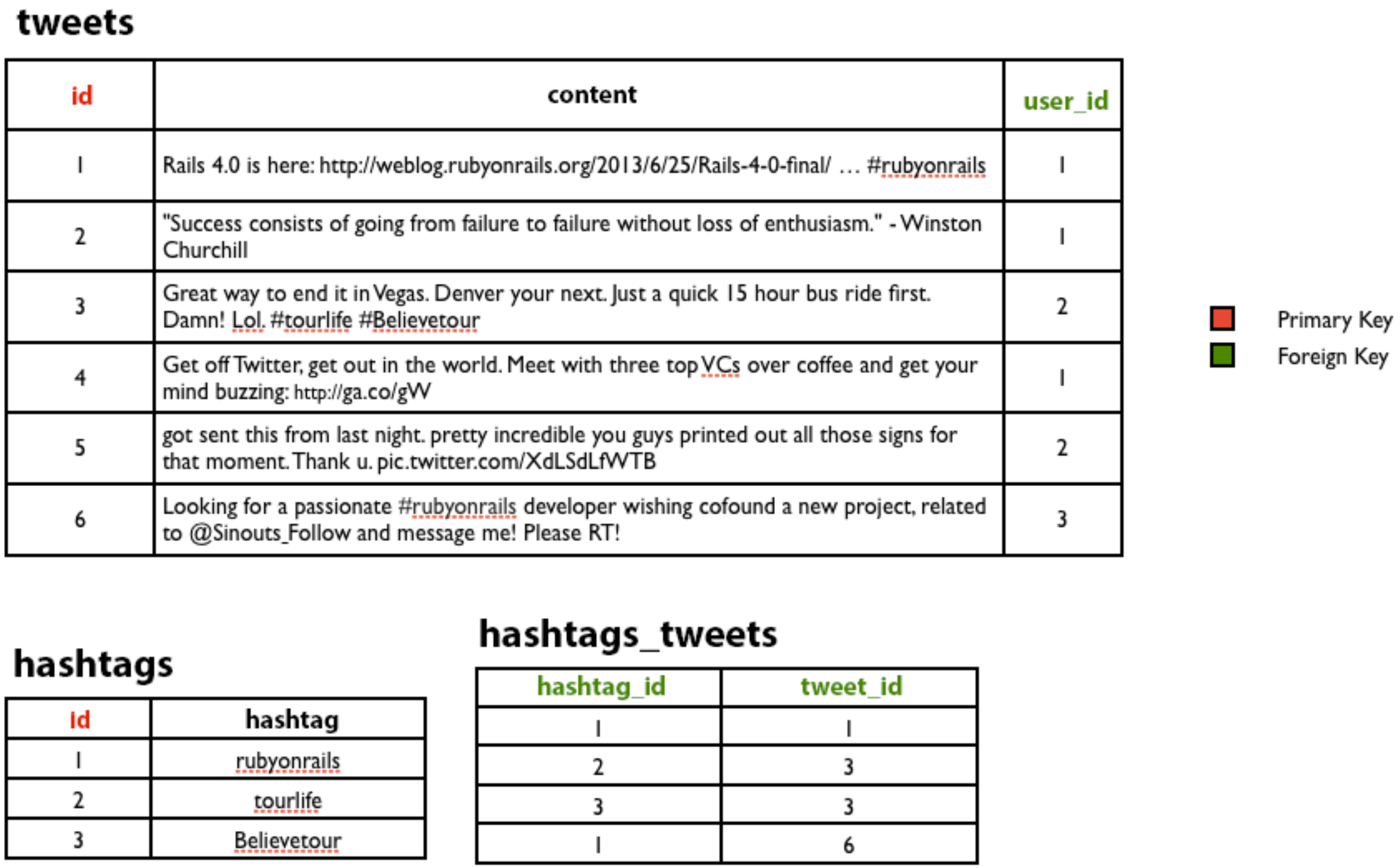
They usually only have two foreign keys.

- `has_and_belongs_to_many`
- `has_many :through`
- `has_one :through`

Tweets and Hashtags are both *has_and_belong_to_many.*
Therefore, we need a join table called *hashtags_tweets* to connect them.

```
rails generate migration CreateHashtagTweetJoinTable hashtags tweets
```

JOIN TABLE : HASHTAGS_TWEETS



tweets

id	content	user_id
1	Rails 4.0 is here: http://weblog.rubyonrails.org/2013/6/25/Rails-4-0-final/ … #rubyonrails	1
2	"Success consists of going from failure to failure without loss of enthusiasm." - Winston Churchill	1
3	Great way to end it in Vegas. Denver your next. Just a quick 15 hour bus ride first. Damn! Lol. #tourlife #Believetour	2
4	Get off Twitter, get out in the world. Meet with three top VCs over coffee and get your mind buzzing: http://ga.co/gW	1
5	got sent this from last night. pretty incredible you guys printed out all those signs for that moment. Thank u. pic.twitter.com/XdLSdLfWTB	2
6	Looking for a passionate #rubyonrails developer wishing cofound a new project, related to @Sinouts_Follow and message me! Please RT!	3

🟥	Primary Key
🟩	Foreign Key

hashtags

id	hashtag
1	rubyonrails
2	tourlife
3	Believetour

hashtags_tweets

hashtag_id	tweet_id
1	1
2	3
3	3
1	6

PAIR UP!

ASSOCIATION & SCHEMA DESIGN: NETFLIX

PAIR UP EXERCISE: NETFLIX!

List the database tables for Netflix!

starter_code/netflix_associations.md

1. Define your resources first
2. Draw connections between them.
3. Determine the type of association

- has_one/belongs_to
- has_many/belongs_to
- has_and_belongs_to_many

RESOURCE:

http://guides.rubyonrails.org/association_basics.htm

HOMEWORK

GAME PLAN & EXPECTATIONS

FINISH PART 1, 2 & 3 OF MUSIC APP

DUE Next Wednesday

MUSIC APP!

CODE ALONG & LAB TIME!