

BEWD 10

SCHEMA DESIGN &  
ASSOCIATIONS

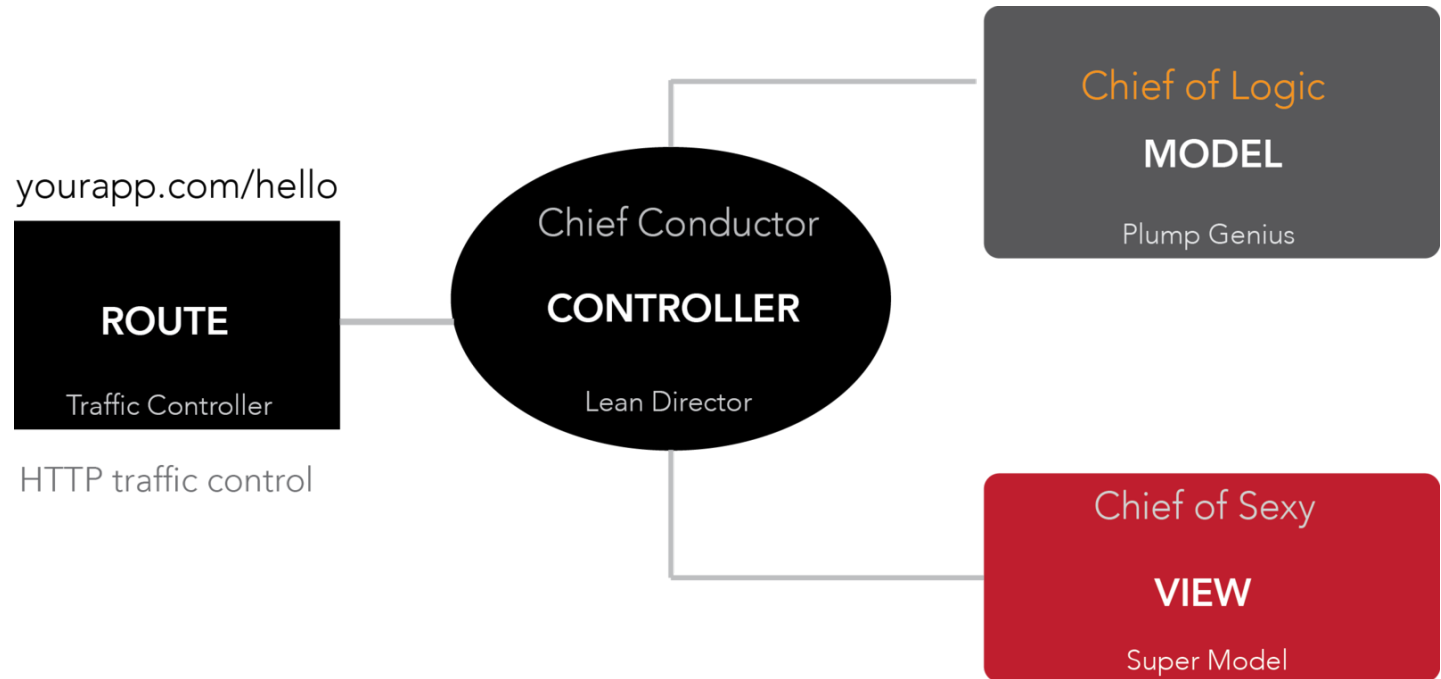
# LEARNING GOALS

## **MVC - MODEL 101**

- HOW SCHEMA DESIGN WORKS
- HOW TO CREATE ASSOCIATIONS

# ASSOCIATIONS

ACTIVE RECORD: MODEL



# 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.

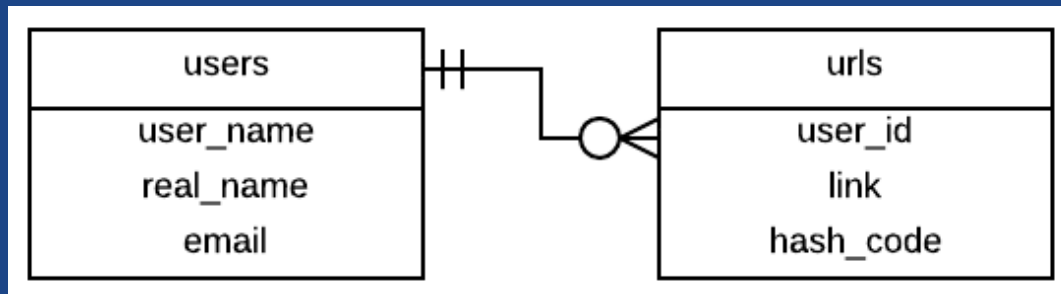


```
create_table "users", force: true do |t|  
  t.string :user_name  
  t.string :real_name  
  t.string :email  
  ...  
end
```

WE CONNECT URLS AND USERS USING AN ASSOCIATION

# ASSOCIATIONS: CONCEPT & CODE

## *AN ASSOCIATION USING AN ENTITY-RELATIONSHIP DIAGRAM*



```
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. IN RAILS, THIS IS ALWAYS THE `:id`
- THE **FOREIGN KEY** IDENTIFIES WHICH ROW OF ANOTHER TABLE IT IS ASSOCIATE WITH `:USER_ID`

```
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**

WHY TIME?

# EXAMPLE!

TWITTER: SCHEMA DESIGN

# SCHEMA DESIGN: TWITTER

## RESOURCES

- **User**
- **Tweet**
- **HashTags**

## ASSOCIATIONS

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

# SCHEMA DESIGN

## User

ID	Handle	Name	Email
1	@GA	General Assembly	<a href="mailto:ga@generalassembly.ly">ga@generalassembly.ly</a>
2	@justinbieber	Justin Bieber	<a href="mailto:jb@thebiebs.com">jb@thebiebs.com</a>
3	@techCofounder	Some Guy	<a href="mailto:someguy@yahoo.com">someguy@yahoo.com</a>

■ Primary Key  
■ Foreign Key

## Tweet

ID	Content	user_id
1	Rails 4.0 is here: <a href="http://weblog.rubyonrails.org/2013/6/25/Rails-4-0-final/">http://weblog.rubyonrails.org/2013/6/25/Rails-4-0-final/</a> ... <a href="#">#rubyonrails</a>	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! <a href="#">LoL</a> . <a href="#">#tourlife</a> <a href="#">#Believetour</a>	2
4	Get off Twitter, get out in the world. Meet with three top <a href="#">VCs</a> over coffee and get your mind buzzing: <a href="http://ga.co/gW">http://ga.co/gW</a>	1
5	got sent this from last night. pretty incredible you guys printed out all those signs for that moment. Thank u. <a href="http://pic.twitter.com/XdLSdLfWTB">pic.twitter.com/XdLSdLfWTB</a>	2
6	Looking for a passionate <a href="#">#rubyonrails</a> developer wishing cofound a new project, related to @Sinouts Follow and message me! Please RT!	3

# SCHEMA DESIGN

```
#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\_belongs\_to\_many*.  
Therefore, we need a join table called *hashtags\_tweets* to connect them.

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

# JOIN TABLE :

## tweets

id	content	user_id
1	Rails 4.0 is here: <a href="http://weblog.rubyonrails.org/2013/6/25/Rails-4-0-final/">http://weblog.rubyonrails.org/2013/6/25/Rails-4-0-final/</a> ... <a href="#">#rubyonrails</a>	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. <a href="#">#tourlife</a> <a href="#">#Believetour</a>	2
4	Get off Twitter, get out in the world. Meet with three top <a href="#">VCs</a> over coffee and get your mind buzzing: <a href="http://ga.co/gVv">http://ga.co/gVv</a>	1
5	got sent this from last night. pretty incredible you guys printed out all those signs for that moment. Thank u. <a href="http://pic.twitter.com/XdLSdLfVWTB">pic.twitter.com/XdLSdLfVWTB</a>	2
6	Looking for a passionate <a href="#">#rubyonrails</a> developer wishing cofound a new project, related to <a href="#">@Sinouts_Follow</a> and message me! Please RT!	3

■ Primary Key  
■ Foreign Key

## hashtags

id	hashtag
1	<a href="#">rubyonrails</a>
2	<a href="#">tourlife</a>
3	<a href="#">Believetour</a>

## hashtags\_tweets

hashtag_id	tweet_id
1	1
2	3
3	3
1	6



# PAIR UP!

ASSOCIATIONS & SCHEMA  
DESIGN WITH **NETFLIX**

# PAIR UP EXERCISE

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](http://guides.rubyonrails.org/association_basics.htm)

# HOMEWORK

## GAME PLAN & EXPECTATIONS

FINISH PART 1, 2 & 3 OF MUSIC APP  
DUE MONDAY

# MUSIC APP!

CODE ALONG & LAB TIME!