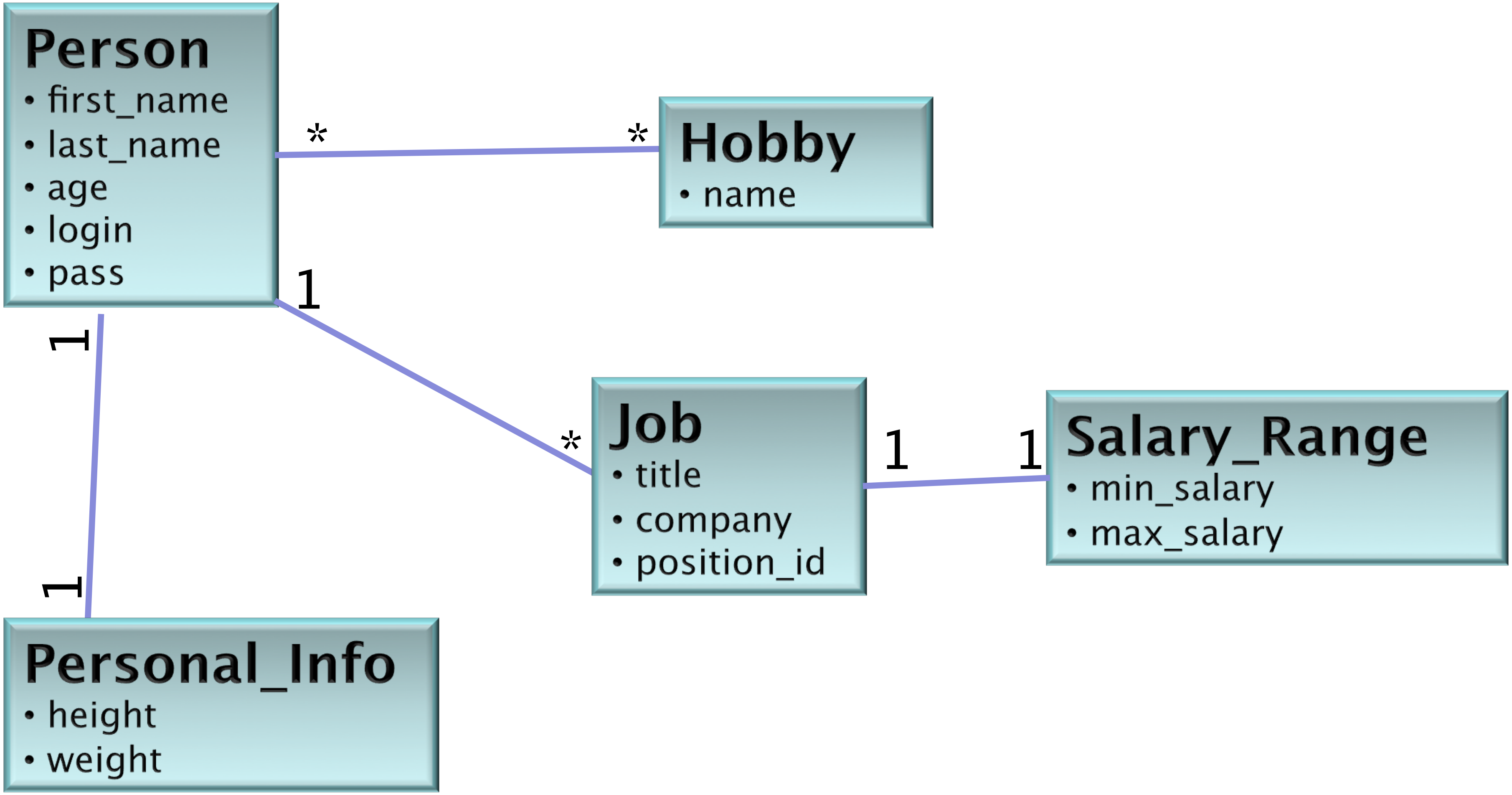
Relationships

Relationships

- ActiveRecord is pretty awesome so far, but what about the relational aspect of the database?
- The individual Ruby models map pretty well to individual tables in the database, but how easy is it to maintain database relationships and build on them?

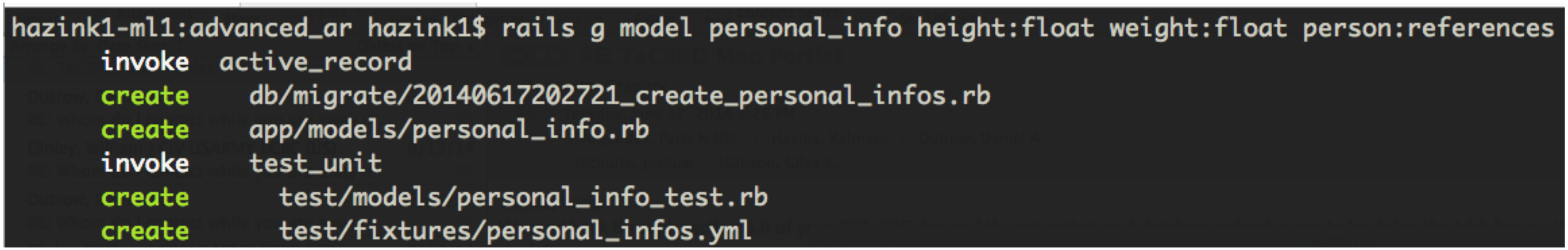
ER diagram


Person
• first_name
• last_name
• age
• login
• pass
*
*
Hobby
• name
1
1
1
*
Job
• title
• company
• position_id
1
1
Salary_Range
• min_salary
• max_salary
Personal_Info
• height
• weight

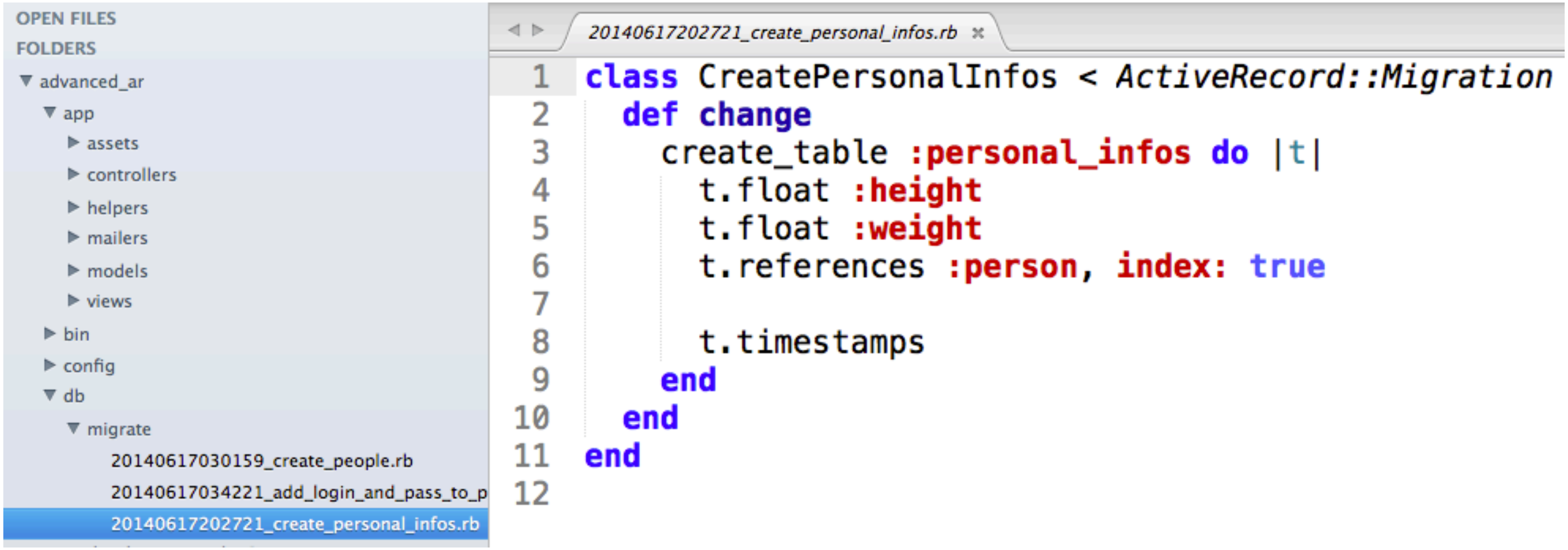
One-to-One Association

- One person *has* exactly *one* personal_info entry
- One personal_info entry *belongs to* exactly one person
- The "*belongs to*" side is the one with a foreign key
- **Convention:** Default name for the foreign key is {master_table_singular}_id, e.g. person_id

One-to-One Association



```
hazink1-ml1:advanced_ar hazink1$ rails g model personal_info height:float weight:float person:references
      invoke  active_record
      create    db/migrate/20140617202721_create_personal_infos.rb
      create    app/models/personal_info.rb
      invoke    test_unit
      create      test/models/personal_info_test.rb
      create      test/fixtures/personal_infos.yml
```

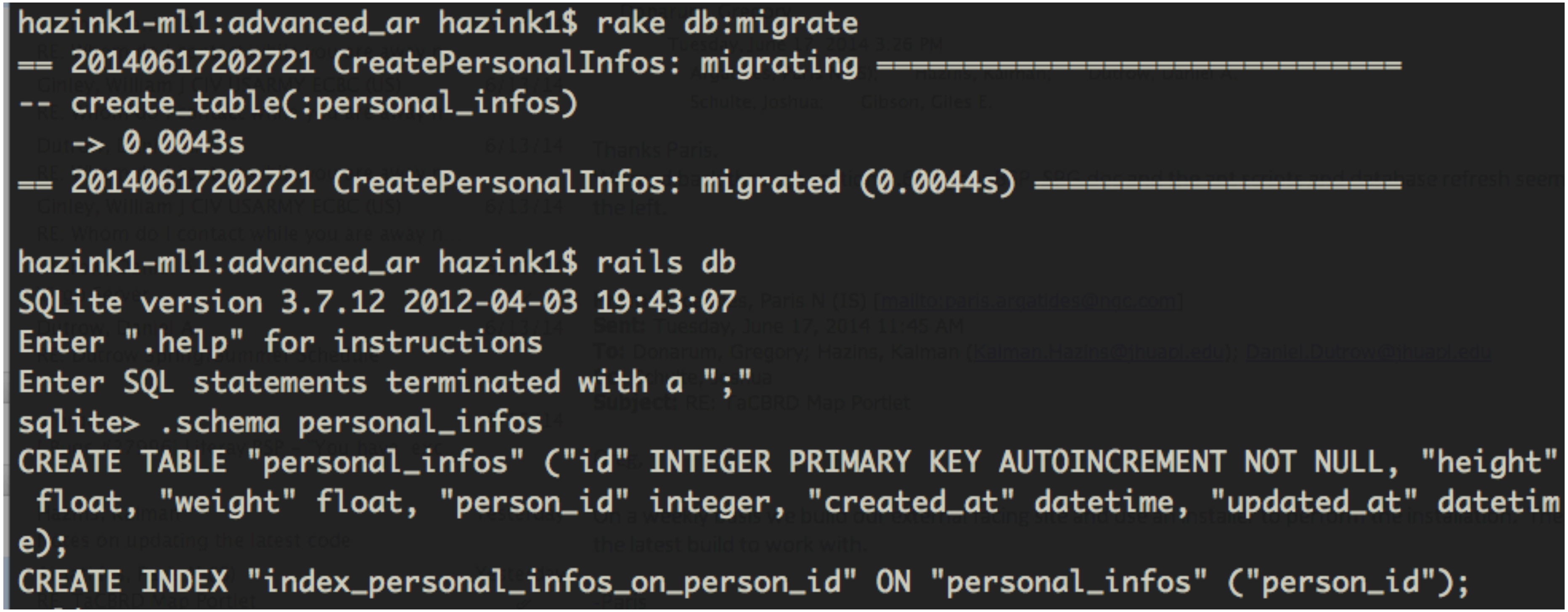


OPEN FILES
FOLDERS
advanced_ar
app
assets
controllers
helpers
mailers
models
views
bin
config
db
migrate
20140617030159_create_people.rb
20140617034221_add_login_and_pass_to_p
20140617202721_create_personal_infos.rb
20140617202721_create_personal_infos.rb

```ruby
class CreatePersonalInfos < ActiveRecord::Migration
  def change
    create_table :personal_infos do |t|
      t.float :height
      t.float :weight
      t.references :person, index: true

      t.timestamps
    end
  end
end

```

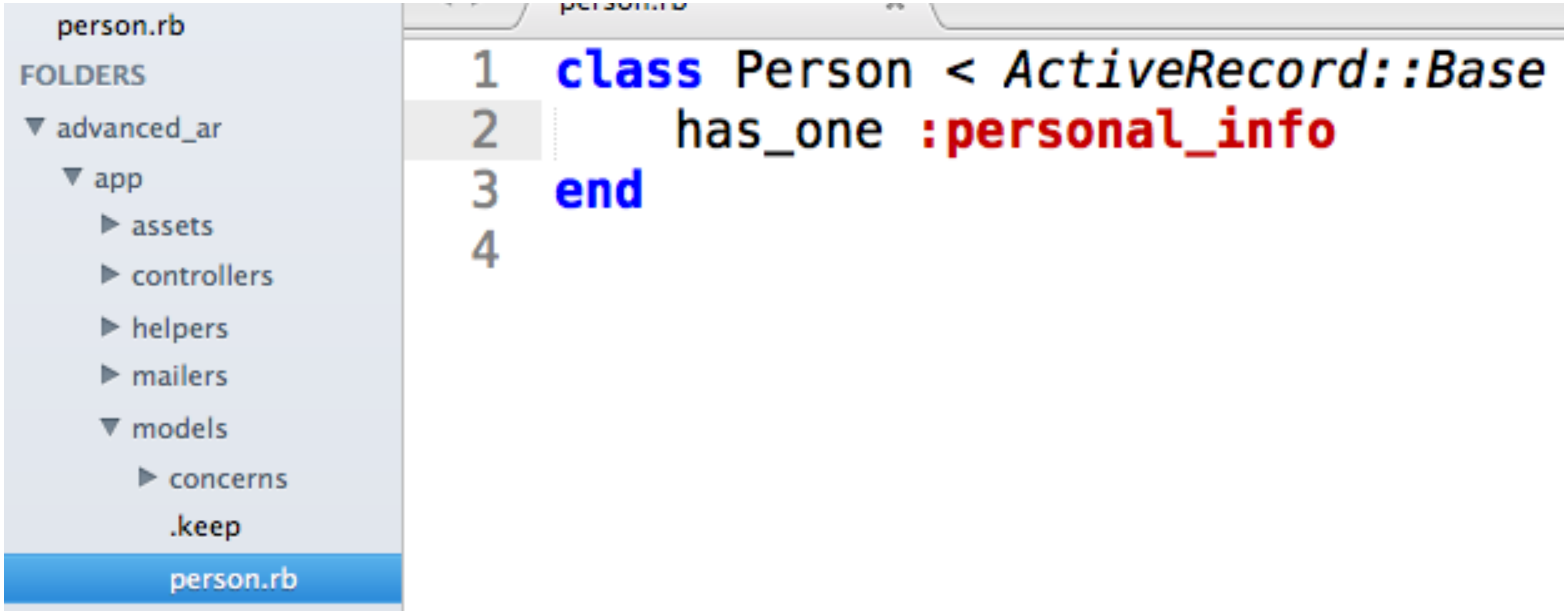
One-to-One Association

Foreign key to people table

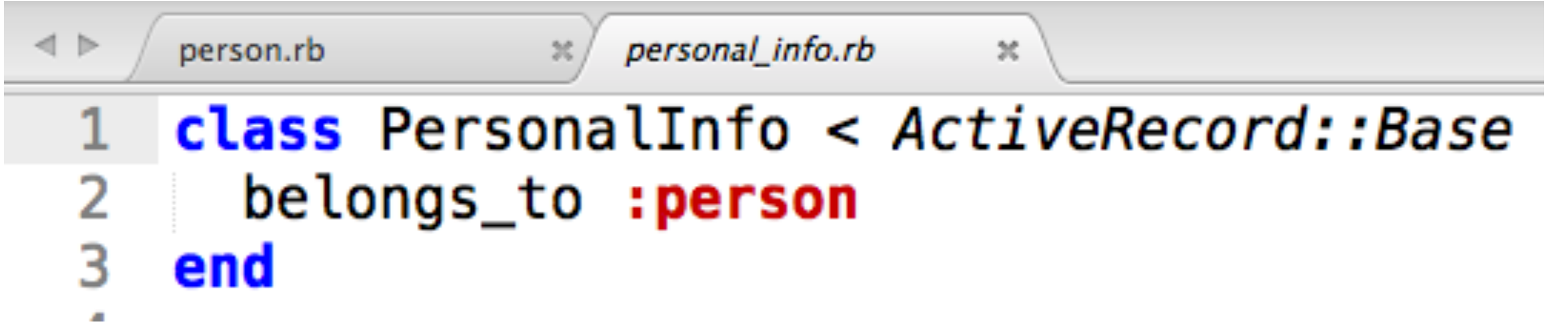


```
hazink1-ml1:advanced_ar hazink1$ rake db:migrate
== 20140617202721 CreatePersonalInfos: migrating ==============================
-- create_table(:personal_infos)
   -> 0.0043s
== 20140617202721 CreatePersonalInfos: migrated (0.0044s) =====================

hazink1-ml1:advanced_ar hazink1$ rails db
SQLite version 3.7.12 2012-04-03 19:43:07
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite> .schema personal_infos
CREATE TABLE "personal_infos" ("id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, "height"
 float, "weight" float, "person_id" integer, "created_at" datetime, "updated_at" datetim
e);
CREATE INDEX "index_personal_infos_on_person_id" ON "personal_infos" ("person_id");
```

One-to-One Association

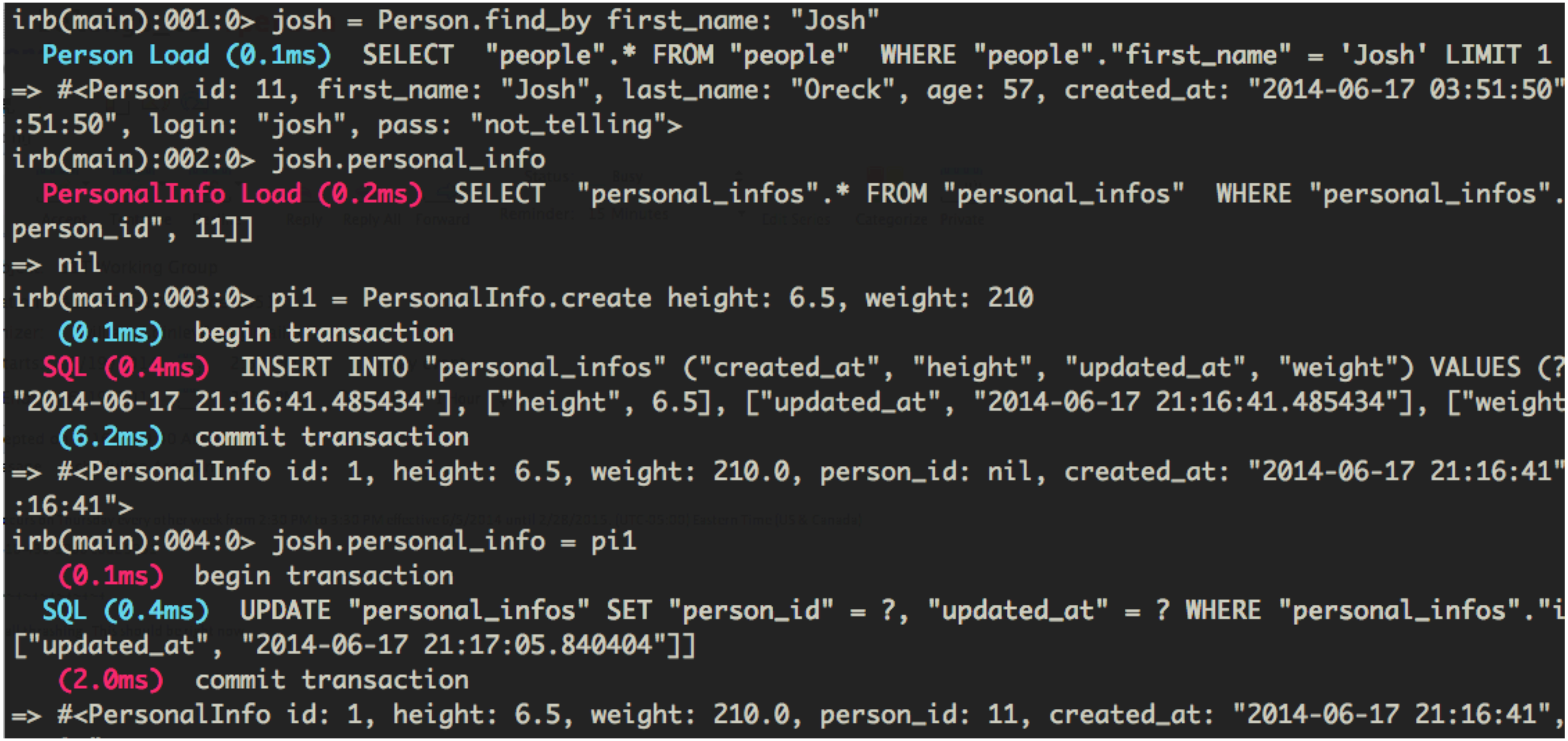


```ruby
class Person < ActiveRecord::Base
    has_one :personal_info
end

```



```ruby
class PersonalInfo < ActiveRecord::Base
  belongs_to :person
end
```

One-to-One Association



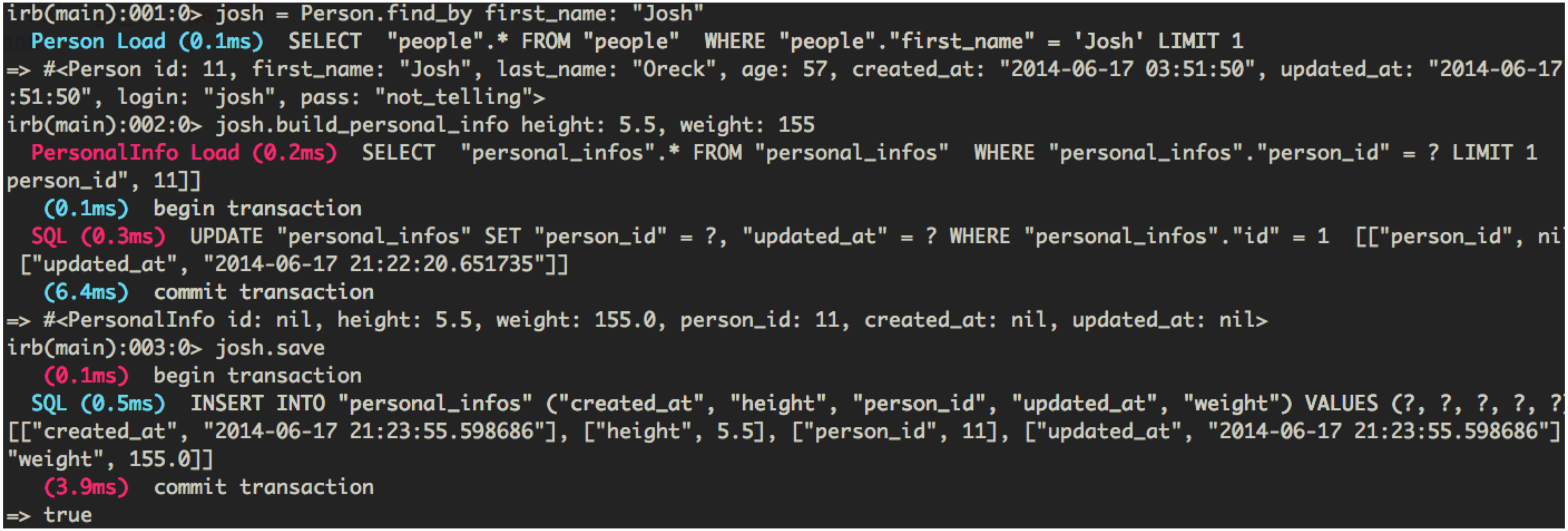
```
irb(main):001:0> josh = Person.find_by first_name: "Josh"
  Person Load (0.1ms)  SELECT  "people".* FROM "people"  WHERE "people"."first_name" = 'Josh' LIMIT 1
=> #<Person id: 11, first_name: "Josh", last_name: "Oreck", age: 57, created_at: "2014-06-17 03:51:50"
:51:50", login: "josh", pass: "not_telling">
irb(main):002:0> josh.personal_info
  PersonalInfo Load (0.2ms)  SELECT  "personal_infos".* FROM "personal_infos"  WHERE "personal_infos".
person_id", 11]]
=> nil
irb(main):003:0> pi1 = PersonalInfo.create height: 6.5, weight: 210
   (0.1ms)  begin transaction
  SQL (0.4ms)  INSERT INTO "personal_infos" ("created_at", "height", "updated_at", "weight") VALUES (?
"2014-06-17 21:16:41.485434"], ["height", 6.5], ["updated_at", "2014-06-17 21:16:41.485434"], ["weight
   (6.2ms)  commit transaction
=> #<PersonalInfo id: 1, height: 6.5, weight: 210.0, person_id: nil, created_at: "2014-06-17 21:16:41"
:16:41">
irb(main):004:0> josh.personal_info = pi1
   (0.1ms)  begin transaction
  SQL (0.4ms)  UPDATE "personal_infos" SET "person_id" = ?, "updated_at" = ? WHERE "personal_infos"."i
["updated_at", "2014-06-17 21:17:05.840404"]]
   (2.0ms)  commit transaction
=> #<PersonalInfo id: 1, height: 6.5, weight: 210.0, person_id: 11, created_at: "2014-06-17 21:16:41",
```

Person and PersonalInfo

Both create PersonalInfo record, but create also creates an entry in the db

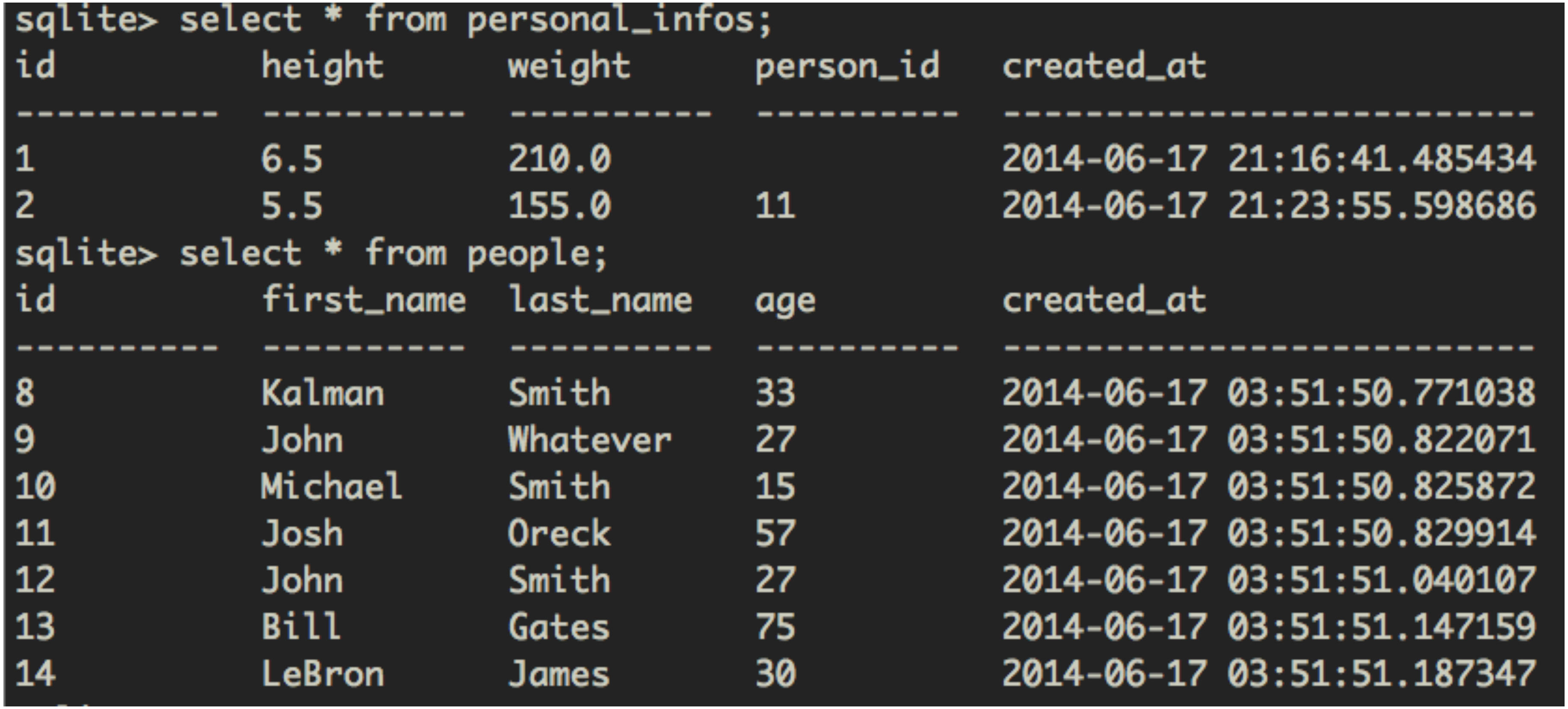
- In addition, you now also have `build_personal_info(hash)` and `create_personal_info(hash)` methods on a person instance
- `create_personal_info` creates a record in the DB right away, while `build_personal_info` does not
- **Both** remove the previous reference in the DB

build_personal_info



```
irb(main):001:0> josh = Person.find_by first_name: "Josh"
  Person Load (0.1ms)  SELECT  "people".* FROM "people"  WHERE "people"."first_name" = 'Josh' LIMIT 1
=> #<Person id: 11, first_name: "Josh", last_name: "Oreck", age: 57, created_at: "2014-06-17 03:51:50", updated_at: "2014-06-17
:51:50", login: "josh", pass: "not_telling">
irb(main):002:0> josh.build_personal_info height: 5.5, weight: 155
  PersonalInfo Load (0.2ms)  SELECT  "personal_infos".* FROM "personal_infos"  WHERE "personal_infos"."person_id" = ? LIMIT 1
person_id", 11]]
   (0.1ms)  begin transaction
  SQL (0.3ms)  UPDATE "personal_infos" SET "person_id" = ?, "updated_at" = ? WHERE "personal_infos"."id" = 1  [["person_id", ni
 ["updated_at", "2014-06-17 21:22:20.651735"]]
   (6.4ms)  commit transaction
=> #<PersonalInfo id: nil, height: 5.5, weight: 155.0, person_id: 11, created_at: nil, updated_at: nil>
irb(main):003:0> josh.save
   (0.1ms)  begin transaction
  SQL (0.5ms)  INSERT INTO "personal_infos" ("created_at", "height", "person_id", "updated_at", "weight") VALUES (?, ?, ?, ?, ?
[["created_at", "2014-06-17 21:23:55.598686"], ["height", 5.5], ["person_id", 11], ["updated_at", "2014-06-17 21:23:55.598686"]
"weight", 155.0]]
   (3.9ms)  commit transaction
=> true
```

people and personal_infos



```
sqlite> select * from personal_infos;
id          height      weight      person_id   created_at
----------  ----------  ----------  ----------  --------------------------
1           6.5         210.0                   2014-06-17 21:16:41.485434
2           5.5         155.0       11          2014-06-17 21:23:55.598686
sqlite> select * from people;
id          first_name  last_name   age         created_at
----------  ----------  ----------  ----------  --------------------------
8           Kalman      Smith       33          2014-06-17 03:51:50.771038
9           John        Whatever    27          2014-06-17 03:51:50.822071
10          Michael     Smith       15          2014-06-17 03:51:50.825872
11          Josh        Oreck       57          2014-06-17 03:51:50.829914
12          John        Smith       27          2014-06-17 03:51:51.040107
13          Bill        Gates       75          2014-06-17 03:51:51.147159
14          LeBron      James       30          2014-06-17 03:51:51.187347
```