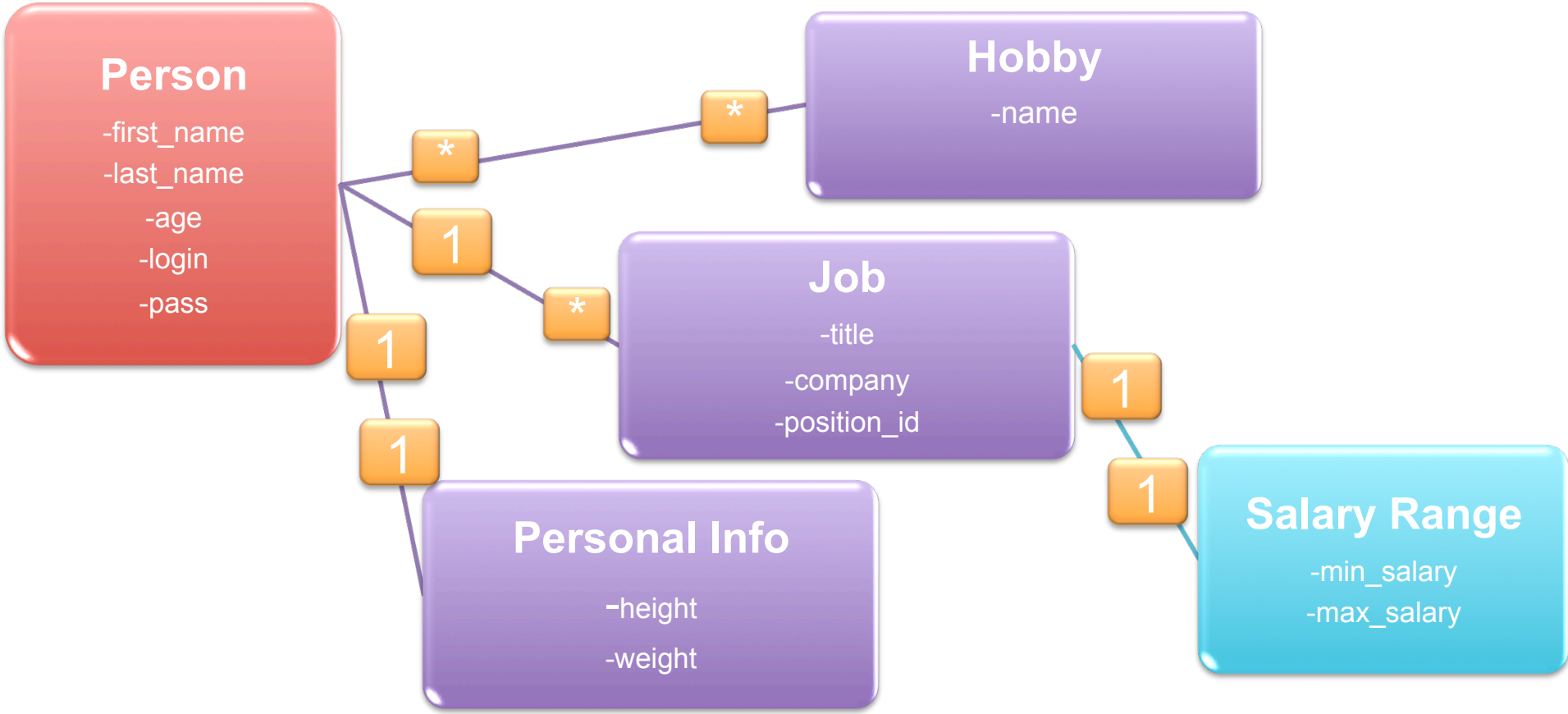
In this lecture, we will discuss…

- ✧ Relational aspect of RDBMS
- ✧ One-to-One Association/Relation

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 relations and build on them?

ER Diagram


Person
-first_name
-last_name
-age
-login
-pass
Hobby
-name
Job
-title
-company
-position_id
Personal Info
-height
-weight
Salary Range
-min_salary
-max_salary
*
*
1
*
1
1
1
1

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

```
~/advanced_ar$ rails g model personal_info height:float weight:float person:references
      invoke  active_record
      create    db/migrate/20150908232650_create_personal_infos.rb
      create    app/models/personal_info.rb
```

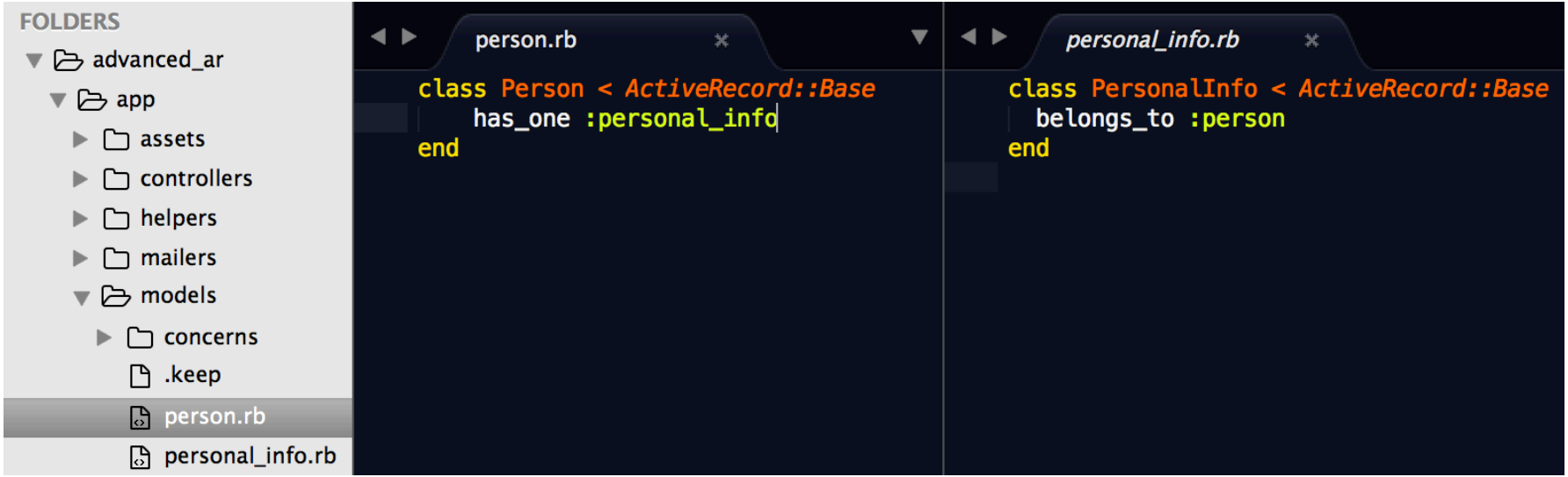
One-to-One Association

```
~/advanced_ar$ rake db:migrate
== 20150908232650 CreatePersonalInfos: migrating ==============================
-- create_table(:personal_infos)
   -> 0.0014s
== 20150908232650 CreatePersonalInfos: migrated (0.0014s) =====================

~/advanced_ar$ rails db
SQLite version 3.8.5 2014-08-15 22:37:57
Enter ".help" for usage hints.
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 NOT NULL, "updated_at" datetime NOT NULL);
CREATE INDEX "index_personal_infos_on_person_id" ON "personal_infos" ("person_id");
```

"Foreign key" to people table

One-to-One Association



One-to-One Association

```
irb(main):001:0> bill = Person.find_by first_name: "Bill"
  Person Load (0.2ms)  SELECT  "people".* FROM "people" WHERE "people"."first_name" = ? LIMIT 1  [["first_name", "Bill"]]
=> #<Person id: 13, first_name: "Bill", age: 75, last_name: "Gates", created_at: "2015-09-08 22:22:51", updated_at: "2015-09-08 22:
22:51", login: "bill", pass: "windows3.1">
irb(main):002:0> bill.personal_info
  PersonalInfo Load (0.1ms)  SELECT  "personal_infos".* FROM "personal_infos" WHERE "personal_infos"."person_id" = ? LIMIT 1  [["pe
rson_id", 13]]
=> nil
irb(main):003:0> pi1 = PersonalInfo.create height: 6.5, weight: 220
   (0.1ms)  begin transaction
  SQL (0.3ms)  INSERT INTO "personal_infos" ("height", "weight", "created_at", "updated_at") VALUES (?, ?, ?, ?)  [["height", 6.5],
 ["weight", 220.0], ["created_at", "2015-09-08 23:39:09.207265"], ["updated_at", "2015-09-08 23:39:09.207265"]]
   (1.4ms)  commit transaction
=> #<PersonalInfo id: 1, height: 6.5, weight: 220.0, person_id: nil, created_at: "2015-09-08 23:39:09", updated_at: "2015-09-08 23:
39:09">
irb(main):004:0> bill.personal_info = pi1
   (0.1ms)  begin transaction
  SQL (0.3ms)  UPDATE "personal_infos" SET "person_id" = ?, "updated_at" = ? WHERE "personal_infos"."id" = ?  [["person_id", 13], [
"updated_at", "2015-09-08 23:39:32.492655"], ["id", 1]]
   (0.7ms)  commit transaction
=> #<PersonalInfo id: 1, height: 6.5, weight: 220.0, person_id: 13, created_at: "2015-09-08 23:39:09", updated_at: "2015-09-08 23:3
9:32">
```

Person and PersonalInfo

- 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 / create_personal_info

```
Loading development environment (Rails 4.2.3)
irb(main):001:0> bill = Person.find_by last_name: "Gates"
  Person Load (0.2ms)  SELECT  "people".* FROM "people" WHERE "people"."last_name" = ? LIMIT 1  [["last_name", "Gates"]]
=> #<Person id: 13, first_name: "Bill", age: 75, last_name: "Gates", created_at: "2015-09-08 22:22:51", updated_at: "2015-09-08 22:22:51", login: "
>
irb(main):002:0> bill.personal_info
  PersonalInfo Load (0.5ms)  SELECT  "personal_infos".* FROM "personal_infos" WHERE "personal_infos"."person_id" = ? LIMIT 1  [["person_id", 13]]
=> #<PersonalInfo id: 1, height: 6.5, weight: 220.0, person_id: 13, created_at: "2015-09-08 23:39:09", updated_at: "2015-09-08 23:39:32">
irb(main):003:0> bill.build_personal_info height: 6.0, weight: 180
   (0.2ms)  begin transaction
  SQL (0.5ms)  UPDATE "personal_infos" SET "person_id" = ?, "updated_at" = ? WHERE "personal_infos"."id" = ?  [["person_id", nil], ["updated_at", "
2"], ["id", 1]]
   (0.7ms)  commit transaction
=> #<PersonalInfo id: nil, height: 6.0, weight: 180.0, person_id: 13, created_at: nil, updated_at: nil>
irb(main):004:0> bill.save
   (0.1ms)  begin transaction
  SQL (1.1ms)  INSERT INTO "personal_infos" ("height", "weight", "person_id", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?)  [["height", 6.0],
son_id", 13], ["created_at", "2015-09-10 23:08:29.192565"], ["updated_at", "2015-09-10 23:08:29.192565"]]
   (1.4ms)  commit transaction
=> true
irb(main):005:0> josh = Person.find_by first_name: "Josh"; josh.create_personal_info height: 5.5, weight: 135
  Person Load (0.3ms)  SELECT  "people".* FROM "people" WHERE "people"."first_name" = ? LIMIT 1  [["first_name", "Josh"]]
   (0.0ms)  begin transaction
  SQL (0.9ms)  INSERT INTO "personal_infos" ("height", "weight", "person_id", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?)  [["height", 5.5],
son_id", 11], ["created_at", "2015-09-10 23:09:36.391913"], ["updated_at", "2015-09-10 23:09:36.391913"]]
   (1.4ms)  commit transaction
  PersonalInfo Load (0.1ms)  SELECT  "personal_infos".* FROM "personal_infos" WHERE "personal_infos"."person_id" = ? LIMIT 1  [["person_id", 11]]
=> #<PersonalInfo id: 3, height: 5.5, weight: 135.0, person_id: 11, created_at: "2015-09-10 23:09:36", updated_at: "2015-09-10 23:09:36">
```

people and personal_infos

```
sqlite> select * from personal_infos;
id          height      weight      person_id   created_at                  updated_at
----------  ----------  ----------  ----------  --------------------------  --------------------------
1           6.5         220.0                   2015-09-08 23:39:09.207265  2015-09-10 23:08:11.687362
2           6.0         180.0       13          2015-09-10 23:08:29.192565  2015-09-10 23:08:29.192565
3           5.5         135.0       11          2015-09-10 23:09:36.391913  2015-09-10 23:09:36.391913
sqlite> select * from people;
id          first_name  age         last_name   created_at                  updated_at                  login       pass
----------  ----------  ----------  ----------  --------------------------  --------------------------  ----------  ----------
8           Kalman      33          Smith       2015-09-08 22:22:51.990586  2015-09-08 22:22:51.990586  kman        abc123
9           John        27          Whatever    2015-09-08 22:22:51.992746  2015-09-08 22:22:51.992746  john1       123abc
10          Michael     15          Smith       2015-09-08 22:22:51.994324  2015-09-08 22:22:51.994324  mike        not_tellin
11          Josh        57          Oreck       2015-09-08 22:22:51.995846  2015-09-08 22:22:51.995846  josh        password1
12          John        27          Smith       2015-09-08 22:22:51.997415  2015-09-08 22:22:51.997415  john2       no_idea
13          Bill        75          Gates       2015-09-08 22:22:51.999069  2015-09-08 22:22:51.999069  bill        windows3.1
14          LeBron      30          James       2015-09-08 22:22:52.000502  2015-09-08 22:22:52.000502  bron        need more
```

Summary

- `has_one` / `belongs_to` (and integer column in DB) is all you need to establish a One-to-One association
- ActiveRecord tries enforcing a One-to-One in the DB!

What's Next?

- One-to-Many Association