Tarea: Ecto One To Many
1. Usando generadores, crear Owner (name, age, email, phone_num).

tania@tania-HP:~/Becarios_bunsan/pet_clinic$ mix phx.gen.html PetOwner Owner owners name:string age:integer email:string phone_num:string
* creating lib/pet_clinic_web/controllers/owner_controller.ex
* creating lib/pet_clinic_web/templates/owner/edit.html.heex
* creating lib/pet_clinic_web/templates/owner/form.html.heex
* creating lib/pet_clinic_web/templates/owner/index.html.heex
* creating lib/pet_clinic_web/templates/owner/new.html.heex
* creating lib/pet_clinic_web/templates/owner/show.html.heex
* creating lib/pet_clinic_web/views/owner_view.ex
* creating test/pet_clinic_web/controllers/owner_controller_test.exs
* creating lib/pet_clinic/pet_owner/owner.ex
* creating priv/repo/migrations/20220420212917_create_owners.exs
* creating lib/pet_clinic/pet_owner.ex
* injecting lib/pet_clinic/pet_owner.ex
* creating test/pet_clinic/pet_owner_test.exs
* injecting test/pet_clinic/pet_owner_test.exs
* creating test/support/fixtures/pet_owner_fixtures.ex
* injecting test/support/fixtures/pet_owner_fixtures.ex

Add the resource to your browser scope in lib/pet_clinic_web/router.ex:

```
resources "/owners", OwnerController
```

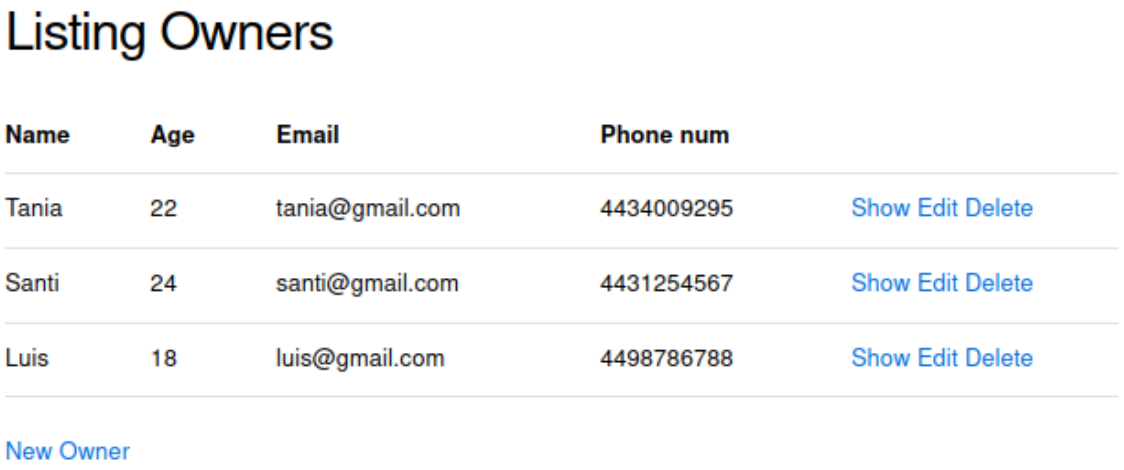
Remember to update your repository by running migrations:

```
$ mix ecto.migrate
```

2. Agregar algunos owners.

Repo.all(Owner)
[debug] QUERY OK source="owners" db=24.2ms decode=10.1ms queue=1.3ms idle=1738.1ms
SELECT o0."id", o0."age", o0."email", o0."name", o0."phone_num", o0."inserted_at", o0."updated_at" FROM "owners" AS o0 []
[
  %PetClinic.PetOwner.Owner{
    __meta__: #Ecto.Schema.Metadata<:loaded, "owners">,
    age: 22,
    email: "tania@gmail.com",
    id: 1,
    inserted_at: ~N[2022-04-20 21:34:23],
    name: "Tania",
    pets: #Ecto.Association.NotLoaded<association :pets is not loaded>,
    phone_num: "4434009295",
    updated_at: ~N[2022-04-20 21:34:23]

```
  },
  %PetClinic.PetOwner.Owner{
        __meta__: #Ecto.Schema.Metadata<:loaded, "owners">,
        age: 24,
        email: "santi@gmail.com",
        id: 2,
        inserted_at: ~N[2022-04-20 21:34:59],
        name: "Santi",
        pets: #Ecto.Association.NotLoaded<association :pets is not loaded>,
        phone_num: "4431254567",
        updated_at: ~N[2022-04-20 21:34:59]
  },
  %PetClinic.PetOwner.Owner{
        __meta__: #Ecto.Schema.Metadata<:loaded, "owners">,
        age: 18,
        email: "luis@gmail.com",
        id: 3,
        inserted_at: ~N[2022-04-20 21:35:36],
        name: "Luis",
        pets: #Ecto.Association.NotLoaded<association :pets is not loaded>,
        phone_num: "4498786788",
        updated_at: ~N[2022-04-20 21:35:36]
  }
]
```



3. Agregar la relación entre Pet y Owner (pet.owner y owner.pets). Modelo y migración.

owner.ex U | pet.ex M | 20220420214807_relate_owners_with_p

lib > pet_clinic > pet_owner > owner.ex > ...

```elixir
defmodule PetClinic.PetOwner.Owner do
  use Ecto.Schema
  import Ecto.Changeset

  schema "owners" do
    field :age, :integer
    field :email, :string
    field :name, :string
    field :phone_num, :string

    has_many :pets, PetClinic.PetClinicService.Pet

    timestamps()
  end
```

wner.ex U | pet.ex M | 20220420214807_relate_owners_wi

pet_clinic > pet_clinic_service > pet.ex > ...

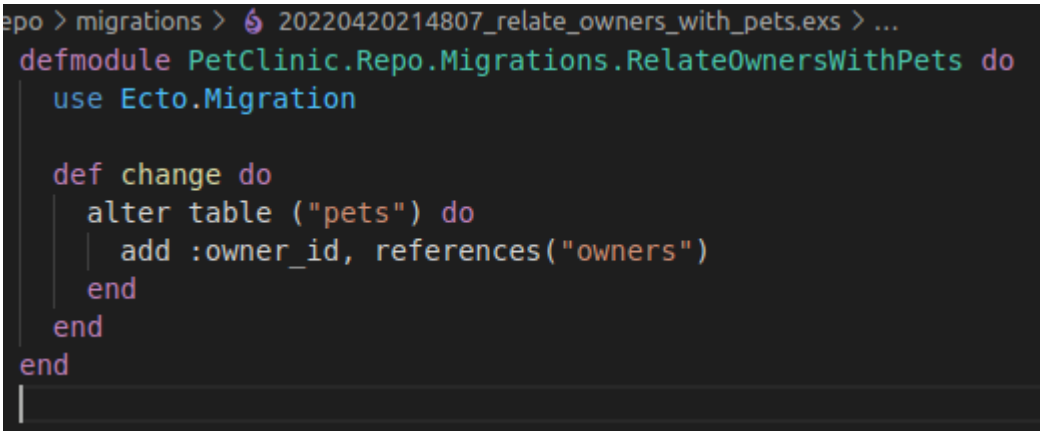
```elixir
defmodule PetClinic.PetClinicService.Pet do
  use Ecto.Schema
  import Ecto.Changeset

  schema "pets" do
    field :age, :integer
    field :name, :string
    field :sex, :string
    field :type, :string

    belongs_to :owner, PetClinic.PetOwner.Owner

    timestamps()
  end
```

mix ecto.gen.migration relate_owners_with_pets
* creating priv/repo/migrations/20220420214807_relate_owners_with_pets.exs



mix ecto.migrate

16:52:42.402 [info] == Running 20220420214807 PetClinic.Repo.Migrations.RelateOwnersWithPets.change/0 forward

16:52:42.414 [info] alter table pets

16:52:42.514 [info] == Migrated 20220420214807 in 0.0s

4. Asociar owners con pets (mandar iex + resultado)

iex(13)> Repo.all(Owner)
[debug] QUERY OK source="owners" db=0.6ms queue=0.1ms idle=1561.0ms
SELECT o0."id", o0."age", o0."email", o0."name", o0."phone_num", o0."inserted_at", o0."updated_at" FROM "owners" AS o0 []
[
  %PetClinic.PetOwner.Owner{
    __meta__: #Ecto.Schema.Metadata<:loaded, "owners">,
    age: 22,
    email: "tania@gmail.com",
    id: 1,
    inserted_at: ~N[2022-04-20 21:34:23],
    name: "Tania",
    pets: #Ecto.Association.NotLoaded<association :pets is not loaded>,
    phone_num: "4434009295",
    updated_at: ~N[2022-04-20 21:34:23]
  },
  %PetClinic.PetOwner.Owner{
    __meta__: #Ecto.Schema.Metadata<:loaded, "owners">,
    age: 24,
    email: "santi@gmail.com",
    id: 2,
    inserted_at: ~N[2022-04-20 21:34:59],
    name: "Santi",
    pets: #Ecto.Association.NotLoaded<association :pets is not loaded>,

```
        phone_num: "4431254567",
        updated_at: ~N[2022-04-20 21:34:59]
  },
  %PetClinic.PetOwner.Owner{
        __meta__: #Ecto.Schema.Metadata<:loaded, "owners">,
        age: 18,
        email: "luis@gmail.com",
        id: 3,
        inserted_at: ~N[2022-04-20 21:35:36],
        name: "Luis",
        pets: #Ecto.Association.NotLoaded<association :pets is not loaded>,
        phone_num: "4498786788",
        updated_at: ~N[2022-04-20 21:35:36]
  }
]
```

iex(15)> Repo.all(Pet)

```
[
  %PetClinic.PetClinicService.Pet{
        __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
        age: 1,
        id: 2,
        inserted_at: ~N[2022-04-06 22:45:08],
        name: "Stormy",
        owner: #Ecto.Association.NotLoaded<association :owner is not loaded>,
        owner_id: nil,
        sex: "female",
        type: "cat",
        updated_at: ~N[2022-04-06 22:45:08]
  },
  %PetClinic.PetClinicService.Pet{
        __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
        age: 2,
        id: 6,
        inserted_at: ~N[2022-04-07 17:08:48],
        name: "fred",
        owner: #Ecto.Association.NotLoaded<association :owner is not loaded>,
        owner_id: nil,
        sex: "male",
        type: "snake",
        updated_at: ~N[2022-04-07 17:08:48]
  },
  %PetClinic.PetClinicService.Pet{
        __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
        age: 3,
        id: 3,
        inserted_at: ~N[2022-04-07 14:38:59],
        name: "clifford",
```

```
    owner: #Ecto.Association.NotLoaded<association :owner is not loaded>,
    owner_id: nil,
    sex: "male",
    type: "dog",
    updated_at: ~N[2022-04-20 21:23:43]
  }
]
[debug] QUERY OK source="pets" db=0.9ms queue=0.1ms idle=1852.3ms
SELECT p0."id", p0."age", p0."name", p0."sex", p0."type", p0."owner_id", p0."inserted_at",
p0."updated_at" FROM "pets" AS p0 []
iex(16)>
```

4.1 Consultar 2 pets con la asociación hacia owner precargada.

```
iex(19)> pet1 = Repo.get_by(Pet, name: "Stormy") |> Repo.preload(:owner)
[debug] QUERY OK source="pets" db=1.8ms queue=0.1ms idle=1320.2ms
SELECT p0."id", p0."age", p0."name", p0."sex", p0."type", p0."owner_id", p0."inserted_at",
p0."updated_at" FROM "pets" AS p0 WHERE (p0."name" = $1) ["Stormy"]
%PetClinic.PetClinicService.Pet{
  __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
  age: 1,
  id: 2,
  inserted_at: ~N[2022-04-06 22:45:08],
  name: "Stormy",
  owner: nil,
  owner_id: nil,
  sex: "female",
  type: "cat",
  updated_at: ~N[2022-04-06 22:45:08]
}

iex(12)> pet2 = Repo.get_by(Pet, name: "clifford") |> Repo.preload(:owner)
[debug] QUERY OK source="pets" db=1.8ms queue=3.3ms idle=1985.2ms
SELECT p0."id", p0."age", p0."name", p0."sex", p0."type", p0."owner_id", p0."inserted_at",
p0."updated_at" FROM "pets" AS p0 WHERE (p0."name" = $1) ["clifford"]
%PetClinic.PetClinicService.Pet{
  __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
  age: 3,
  id: 3,
  inserted_at: ~N[2022-04-07 14:38:59],
  name: "clifford",
  owner: nil,
  owner_id: nil,
  sex: "male",
  type: "dog",
  updated_at: ~N[2022-04-20 21:23:43]
}
```

4.2 Usando put_assoc, asociar c/u de esos pets con algún owner.

iex(21)> ch1 = pet1 |> change() |> put_assoc(:owner, tania)

```
#Ecto.Changeset<
  action: nil,
  changes: %{
          owner: #Ecto.Changeset<action: :update,
          changes: %{}, errors: [],
          data: #PetClinic.PetOwner.Owner<>, valid?: true>
  },
  errors: [],
  data: #PetClinic.PetClinicService.Pet<>,
  valid?: true
>
```

iex(22)> Repo.update(ch1)

```
[debug] QUERY OK db=0.4ms queue=0.1ms idle=1335.0ms
begin []
[debug] QUERY OK db=1.1ms
UPDATE "pets" SET "owner_id" = $1, "updated_at" = $2 WHERE "id" = $3 [1,
~N[2022-04-20 22:49:07], 2]
[debug] QUERY OK db=5.9ms
commit []
{:ok,
 %PetClinic.PetClinicService.Pet{
   __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
   age: 1,
   id: 2,
   inserted_at: ~N[2022-04-06 22:45:08],
   name: "Stormy",
   owner: %PetClinic.PetOwner.Owner{
          __meta__: #Ecto.Schema.Metadata<:loaded, "owners">,
          age: 22,
          email: "tania@gmail.com",
          id: 1,
          inserted_at: ~N[2022-04-20 21:34:23],
          name: "Tania",
          pets: #Ecto.Association.NotLoaded<association :pets is not loaded>,
          phone_num: "4434009295",
          updated_at: ~N[2022-04-20 21:34:23]
   },
   owner_id: 1,
   sex: "female",
   type: "cat",
   updated_at: ~N[2022-04-20 22:49:07]
 }}
```

iex(13)> ch2 = pet2 |> change() |> put_assoc(:owner, santi)

```
#Ecto.Changeset<
  action: nil,
  changes: %{
          owner: #Ecto.Changeset<action: :update, changes: %{}, errors: [],
          data: #PetClinic.PetOwner.Owner<>, valid?: true>
  },
  errors: [],
  data: #PetClinic.PetClinicService.Pet<>,
  valid?: true
>
iex(14)> Repo.update(ch2)
[debug] QUERY OK db=0.5ms queue=0.1ms idle=1187.7ms
begin []
[debug] QUERY OK db=2.1ms
UPDATE "pets" SET "owner_id" = $1, "updated_at" = $2 WHERE "id" = $3 [2,
~N[2022-04-20 23:06:47], 3]
[debug] QUERY OK db=14.3ms
commit []
{:ok,
 %PetClinic.PetClinicService.Pet{
   __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
   age: 3,
   id: 3,
   inserted_at: ~N[2022-04-07 14:38:59],
   name: "clifford",
   owner: %PetClinic.PetOwner.Owner{
          __meta__: #Ecto.Schema.Metadata<:loaded, "owners">,
          age: 24,
          email: "santi@gmail.com",
          id: 2,
          inserted_at: ~N[2022-04-20 21:34:59],
          name: "Santi",
          pets: #Ecto.Association.NotLoaded<association :pets is not loaded>,
          phone_num: "4431254567",
          updated_at: ~N[2022-04-20 21:34:59]
   },
   owner_id: 2,
   sex: "male",
   type: "dog",
   updated_at: ~N[2022-04-20 23:06:47]
 }}
iex(15)>
```

4.3 Consultar el owner anterior, precargando la asociación con pets.

```
iex(9)> Repo.get_by(Owner, name: "Tania") |> Repo.preload(:pets)
[debug] QUERY OK source="owners" db=2.8ms queue=0.1ms idle=1396.7ms
```

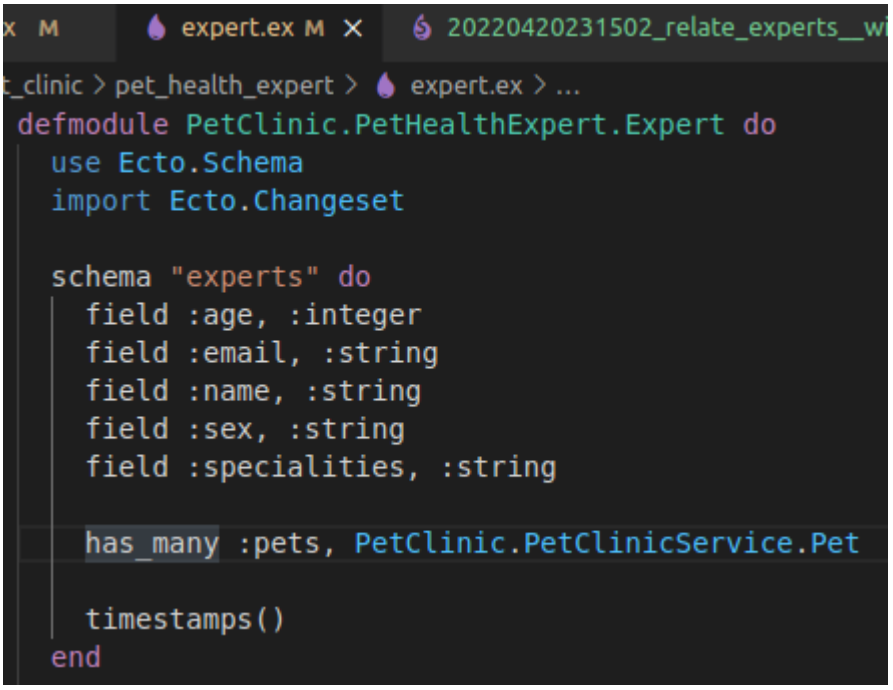
```
SELECT o0."id", o0."age", o0."email", o0."name", o0."phone_num", o0."inserted_at",
o0."updated_at" FROM "owners" AS o0 WHERE (o0."name" = $1) ["Tania"]
[debug] QUERY OK source="pets" db=1.3ms queue=2.6ms idle=1432.5ms
SELECT p0."id", p0."age", p0."name", p0."sex", p0."type", p0."owner_id", p0."inserted_at",
p0."updated_at", p0."owner_id" FROM "pets" AS p0 WHERE (p0."owner_id" = $1) ORDER
BY p0."owner_id" [1]
%PetClinic.PetOwner.Owner{
  __meta__: #Ecto.Schema.Metadata<:loaded, "owners">,
  age: 22,
  email: "tania@gmail.com",
  id: 1,
  inserted_at: ~N[2022-04-20 21:34:23],
  name: "Tania",
  pets: [
        %PetClinic.PetClinicService.Pet{
        __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
        age: 1,
        id: 2,
        inserted_at: ~N[2022-04-06 22:45:08],
        name: "Stormy",
        owner: #Ecto.Association.NotLoaded<association :owner is not loaded>,
        owner_id: 1,
        sex: "female",
        type: "cat",
        updated_at: ~N[2022-04-20 22:49:07]
        }
  ],
  phone_num: "4434009295",
  updated_at: ~N[2022-04-20 21:34:23]
}
```

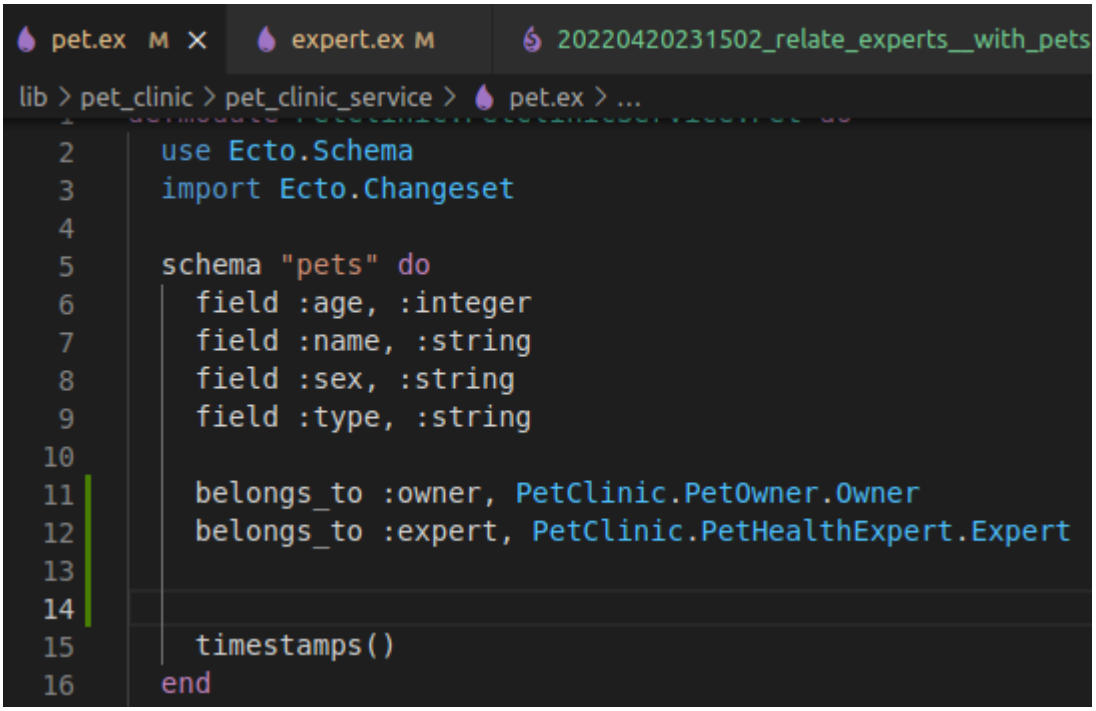
iex(16)> Repo.get_by(Owner, name: "Santi") |> Repo.preload(:pets)

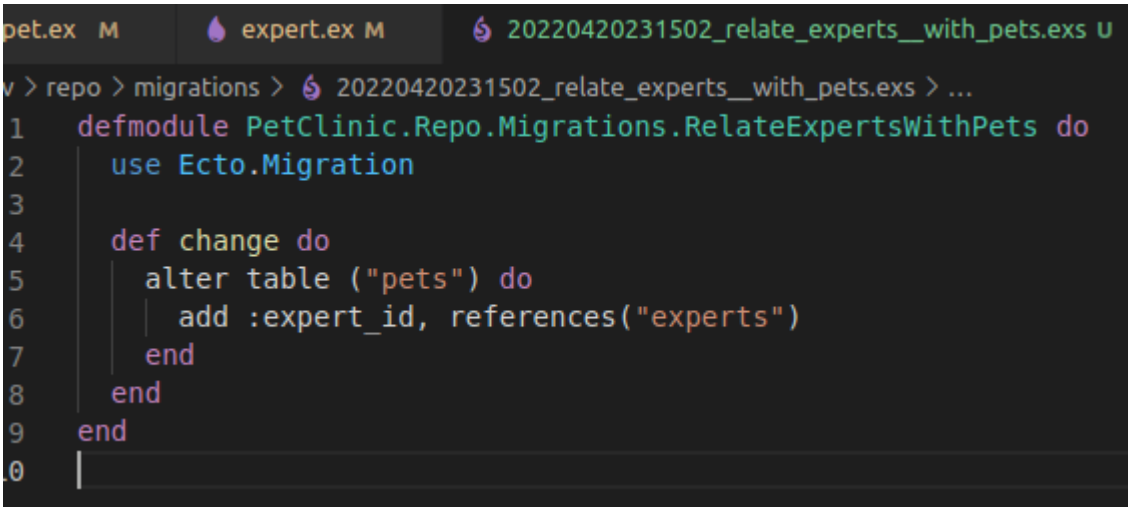
```
[debug] QUERY OK source="owners" db=1.2ms queue=0.1ms idle=1584.5ms
SELECT o0."id", o0."age", o0."email", o0."name", o0."phone_num", o0."inserted_at",
o0."updated_at" FROM "owners" AS o0 WHERE (o0."name" = $1) ["Santi"]
[debug] QUERY OK source="pets" db=7.0ms queue=0.1ms idle=1586.4ms
SELECT p0."id", p0."age", p0."name", p0."sex", p0."type", p0."owner_id", p0."inserted_at",
p0."updated_at", p0."owner_id" FROM "pets" AS p0 WHERE (p0."owner_id" = $1) ORDER
BY p0."owner_id" [2]
%PetClinic.PetOwner.Owner{
  __meta__: #Ecto.Schema.Metadata<:loaded, "owners">,
  age: 24,
  email: "santi@gmail.com",
  id: 2,
  inserted_at: ~N[2022-04-20 21:34:59],
  name: "Santi",
  pets: [
        %PetClinic.PetClinicService.Pet{
```

```
        __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
        age: 3,
        id: 3,
        inserted_at: ~N[2022-04-07 14:38:59],
        name: "clifford",
        owner: #Ecto.Association.NotLoaded<association :owner is not loaded>,
        owner_id: 2,
        sex: "male",
        type: "dog",
        updated_at: ~N[2022-04-20 23:06:47]
        }
  ],
  phone_num: "4431254567",
  updated_at: ~N[2022-04-20 21:34:59]
}
```

5. Repetir las instrucciones del punto 3, pero para Pet y HelthExpert (healthexpert.patients, pet.preferred_expert)







tania@tania-HP:~/Becarios_bunsan/pet_clinic$ mix ecto.gen.migration relate_experts__with_pets
* creating priv/repo/migrations/20220420231502_relate_experts__with_pets.exs
tania@tania-HP:~/Becarios_bunsan/pet_clinic$ mix ecto.migrate

18:17:08.393 [info] == Running 20220420231502 PetClinic.Repo.Migrations.RelateExpertsWithPets.change/0 forward

18:17:08.403 [info] alter table pets

18:17:08.414 [info] == Migrated 20220420231502 in 0.0s
6. Repetir el punto 4 pero para Pet y HelthExpert
6.0
iex(8)> Repo.all(Pet)
[debug] QUERY OK source="pets" db=1.4ms decode=0.1ms queue=2.2ms idle=684.0ms

```
SELECT p0."id", p0."age", p0."name", p0."sex", p0."type", p0."owner_id", p0."expert_id",
p0."inserted_at", p0."updated_at" FROM "pets" AS p0 []
[
  %PetClinic.PetClinicService.Pet{
        __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
        age: 2,
        expert: #Ecto.Association.NotLoaded<association :expert is not loaded>,
        expert_id: nil,
        id: 6,
        inserted_at: ~N[2022-04-07 17:08:48],
        name: "fred",
        owner: #Ecto.Association.NotLoaded<association :owner is not loaded>,
        owner_id: nil,
        sex: "male",
        type: "snake",
        updated_at: ~N[2022-04-07 17:08:48]
  },
  %PetClinic.PetClinicService.Pet{
        __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
        age: 1,
        expert: #Ecto.Association.NotLoaded<association :expert is not loaded>,
        expert_id: nil,
        id: 2,
        inserted_at: ~N[2022-04-06 22:45:08],
        name: "Stormy",
        owner: #Ecto.Association.NotLoaded<association :owner is not loaded>,
        owner_id: 1,
        sex: "female",
        type: "cat",
        updated_at: ~N[2022-04-20 22:49:07]
  },
  %PetClinic.PetClinicService.Pet{
        __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
        age: 3,
        expert: #Ecto.Association.NotLoaded<association :expert is not loaded>,
        expert_id: nil,
        id: 3,
        inserted_at: ~N[2022-04-07 14:38:59],
        name: "clifford",
        owner: #Ecto.Association.NotLoaded<association :owner is not loaded>,
        owner_id: 2,
        sex: "male",
        type: "dog",
        updated_at: ~N[2022-04-20 23:06:47]
  }
]
iex(9)> Repo.all(Expert)
[debug] QUERY OK source="experts" db=0.7ms idle=1917.5ms
```

```
SELECT e0."id", e0."age", e0."email", e0."name", e0."sex", e0."specialities",
e0."inserted_at", e0."updated_at" FROM "experts" AS e0 []
[
  %PetClinic.PetHealthExpert.Expert{
        __meta__: #Ecto.Schema.Metadata<:loaded, "experts">,
        age: 26,
        email: "amir@bunsan.io",
        id: 1,
        inserted_at: ~N[2022-04-06 23:05:07],
        name: "Amir",
        pets: #Ecto.Association.NotLoaded<association :pets is not loaded>,
        sex: "man",
        specialities: "cats, dogs, horses",
        updated_at: ~N[2022-04-06 23:05:07]
  },
  %PetClinic.PetHealthExpert.Expert{
        __meta__: #Ecto.Schema.Metadata<:loaded, "experts">,
        age: 23,
        email: "erick@bunsan.io",
        id: 2,
        inserted_at: ~N[2022-04-06 23:06:20],
        name: "Erick",
        pets: #Ecto.Association.NotLoaded<association :pets is not loaded>,
        sex: "man",
        specialities: "cats, dogs, bears",
        updated_at: ~N[2022-04-06 23:06:20]
  },
  %PetClinic.PetHealthExpert.Expert{
        __meta__: #Ecto.Schema.Metadata<:loaded, "experts">,
        age: 24,
        email: "regina@bunsan.io",
        id: 3,
        inserted_at: ~N[2022-04-06 23:07:13],
        name: "Regina ",
        pets: #Ecto.Association.NotLoaded<association :pets is not loaded>,
        sex: "woman",
        specialities: "dogs, horses, ducks",
        updated_at: ~N[2022-04-06 23:07:13]
  }
]
iex(10)>
```

6.1

iex(11)> pet1 = Repo.get_by(Pet, name: "fred") |> Repo.preload(:expert)

```
[debug] QUERY OK source="pets" db=0.9ms queue=1.8ms idle=1137.4ms
SELECT p0."id", p0."age", p0."name", p0."sex", p0."type", p0."owner_id", p0."expert_id",
p0."inserted_at", p0."updated_at" FROM "pets" AS p0 WHERE (p0."name" = $1) ["fred"]
%PetClinic.PetClinicService.Pet{
```

```
  __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
  age: 2,
  expert: nil,
  expert_id: nil,
  id: 6,
  inserted_at: ~N[2022-04-07 17:08:48],
  name: "fred",
  owner: #Ecto.Association.NotLoaded<association :owner is not loaded>,
  owner_id: nil,
  sex: "male",
  type: "snake",
  updated_at: ~N[2022-04-07 17:08:48]
}
6.2
iex(17)> ch1 = pet1 |> change() |> put_assoc(:expert, regina)
#Ecto.Changeset<
  action: nil,
  changes: %{
          expert: #Ecto.Changeset<action: :update,
          changes: %{}, errors: [],
          data: #PetClinic.PetHealthExpert.Expert<>,
          valid?: true>
  },
  errors: [],
  data: #PetClinic.PetClinicService.Pet<>,
  valid?: true
>
iex(18)> Repo.update(ch1)
[debug] QUERY OK db=0.8ms queue=0.1ms idle=1105.4ms
begin []
[debug] QUERY OK db=1.2ms
UPDATE "pets" SET "expert_id" = $1, "updated_at" = $2 WHERE "id" = $3 [3,
~N[2022-04-20 23:33:03], 6]
[debug] QUERY OK db=23.6ms
commit []
{:ok,
 %PetClinic.PetClinicService.Pet{
   __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
   age: 2,
   expert: %PetClinic.PetHealthExpert.Expert{
          __meta__: #Ecto.Schema.Metadata<:loaded, "experts">,
          age: 24,
          email: "regina@bunsan.io",
          id: 3,
          inserted_at: ~N[2022-04-06 23:07:13],
          name: "Regina",
          pets: #Ecto.Association.NotLoaded<association :pets is not loaded>,
          sex: "woman",
```

```
        specialities: "dogs, horses, ducks",
        updated_at: ~N[2022-04-20 23:30:19]
  },
  expert_id: 3,
  id: 6,
  inserted_at: ~N[2022-04-07 17:08:48],
  name: "fred",
  owner: #Ecto.Association.NotLoaded<association :owner is not loaded>,
  owner_id: nil,
  sex: "male",
  type: "snake",
  updated_at: ~N[2022-04-20 23:33:03]
 }}
```

6.3

iex(19)> Repo.get_by(Expert, name: "Regina") |> Repo.preload(:pets)

```
[debug] QUERY OK source="experts" db=1.0ms queue=0.4ms idle=1363.3ms
SELECT e0."id", e0."age", e0."email", e0."name", e0."sex", e0."specialities",
e0."inserted_at", e0."updated_at" FROM "experts" AS e0 WHERE (e0."name" = $1)
["Regina"]
[debug] QUERY OK source="pets" db=1.1ms queue=1.9ms idle=1373.4ms
SELECT p0."id", p0."age", p0."name", p0."sex", p0."type", p0."owner_id", p0."expert_id",
p0."inserted_at", p0."updated_at", p0."expert_id" FROM "pets" AS p0 WHERE
(p0."expert_id" = $1) ORDER BY p0."expert_id" [3]
%PetClinic.PetHealthExpert.Expert{
  __meta__: #Ecto.Schema.Metadata<:loaded, "experts">,
  age: 24,
  email: "regina@bunsan.io",
  id: 3,
  inserted_at: ~N[2022-04-06 23:07:13],
  name: "Regina",
  pets: [
        %PetClinic.PetClinicService.Pet{
        __meta__: #Ecto.Schema.Metadata<:loaded, "pets">,
        age: 2,
        expert: #Ecto.Association.NotLoaded<association :expert is not loaded>,
        expert_id: 3,
        id: 6,
        inserted_at: ~N[2022-04-07 17:08:48],
        name: "fred",
        owner: #Ecto.Association.NotLoaded<association :owner is not loaded>,
        owner_id: nil,
        sex: "male",
        type: "snake",
        updated_at: ~N[2022-04-20 23:33:03]
        }
  ],
  sex: "woman",
```

```
  specialities: "dogs, horses, ducks",
  updated_at: ~N[2022-04-20 23:30:19]
}
```

7. Leer la documentación de Ecto.Changeset. La parte principal y las funciones cast y change.

8. En Pet.changeset agregar 1 validación para que la edad no sea menor a 0.