elixir

Ecto in Elixir:
Exploring from the basics.

Ruben Amortegui
@ramortegui
https://www.rubenamortegui.com
https://github.com/ramortegui


parademem

What’s new on Elixir

- Production release 1.8.1
  - Custom struct inspections
    - @derive {Inspect, only: []}
    - @derive {Inspect, except: []}
  - Calendar and Time zone functionality
  - Faster compilation time 5% faster on average.
  - Improved instrumentation $callers and $ancestors when working with processes.
  - Enhacements: Eex, ExUnit
  - https://github.com/elixir-lang/elixir/releases

Agenda

- What is Ecto
- Use cases
- Package Structure
- Main Components

What is?

- Integrated query language for Elixir
- Toolkit for data mapping

History

- Based on: Language Integrated Query (LINQ)

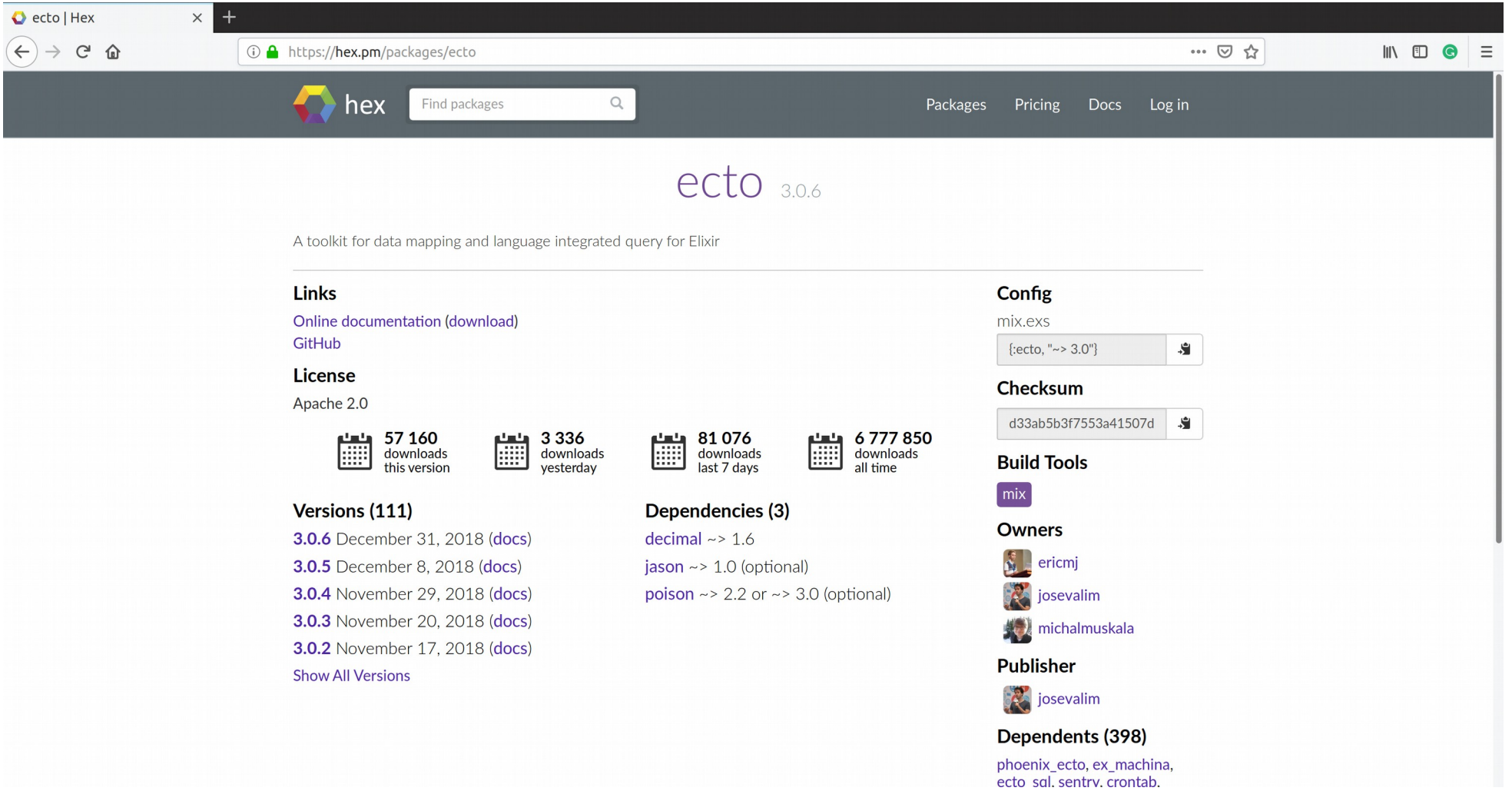
  https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/

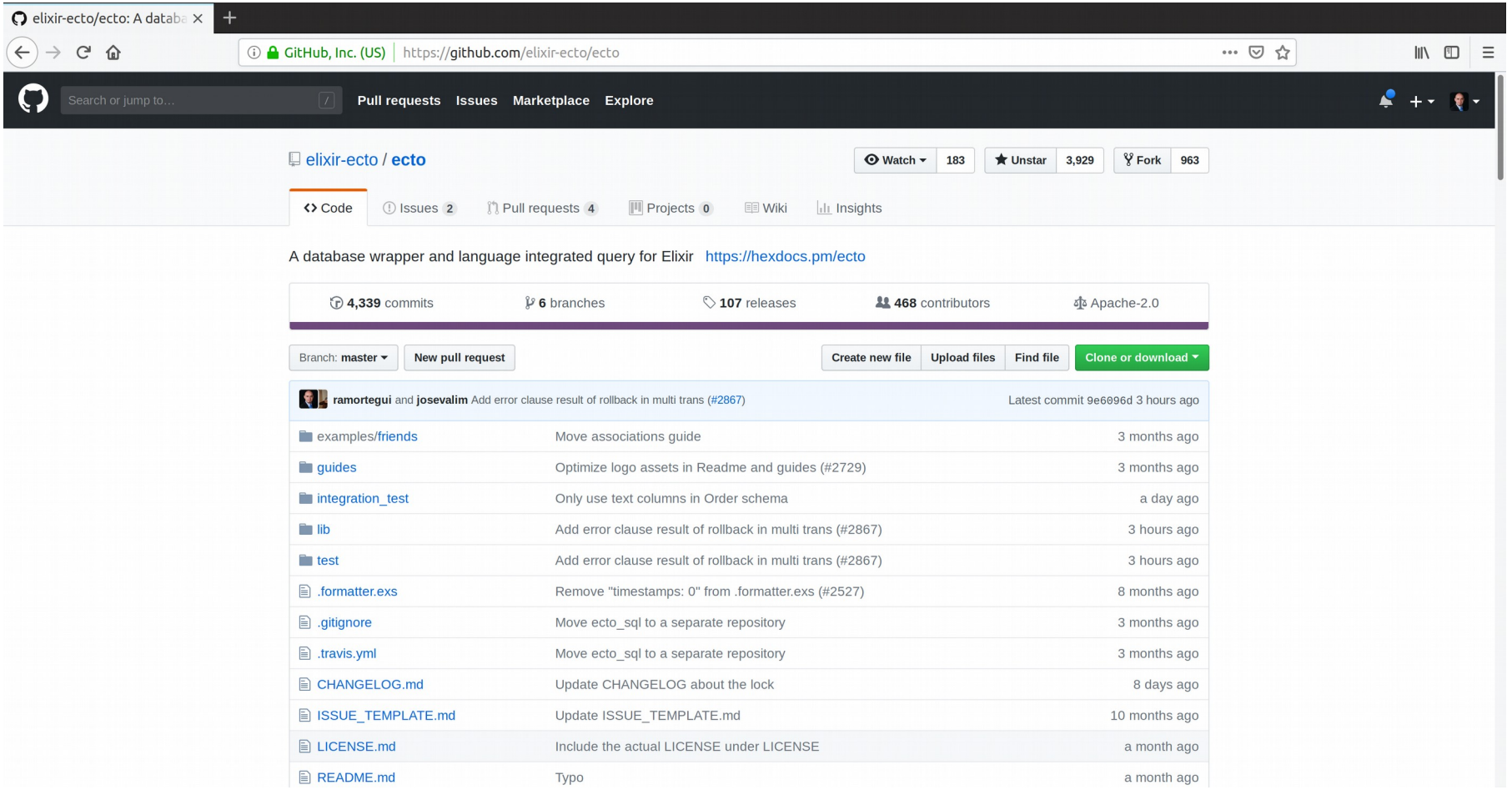
  Released 2007

State / Jan 27th 2019

- Version 3.0.6
- 108 releases
- 474 contributors
- 958 forks
- 3 issues
- 3 pull requests

Package Info




elixir-ecto/ecto: A databa
GitHub, Inc. (US)
https://github.com/elixir-ecto/ecto
Search or jump to…
Pull requests
Issues
Marketplace
Explore
elixir-ecto / ecto
Watch 183
Unstar 3,929
Fork 963
Code
Issues 2
Pull requests 4
Projects 0
Wiki
Insights
A database wrapper and language integrated query for Elixir https://hexdocs.pm/ecto
4,339 commits
6 branches
107 releases
468 contributors
Apache-2.0
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
ramortegui and josevalim Add error clause result of rollback in multi trans (#2867)
Latest commit 9e6096d 3 hours ago
examples/friends Move associations guide 3 months ago
guides Optimize logo assets in Readme and guides (#2729) 3 months ago
integration_test Only use text columns in Order schema a day ago
lib Add error clause result of rollback in multi trans (#2867) 3 hours ago
test Add error clause result of rollback in multi trans (#2867) 3 hours ago
.formatter.exs Remove "timestamps: 0" from .formatter.exs (#2527) 8 months ago
.gitignore Move ecto_sql to a separate repository 3 months ago
.travis.yml Move ecto_sql to a separate repository 3 months ago
CHANGELOG.md Update CHANGELOG about the lock 8 days ago
ISSUE_TEMPLATE.md Update ISSUE_TEMPLATE.md 10 months ago
LICENSE.md Include the actual LICENSE under LICENSE a month ago
README.md Typo a month ago

How Ecto is used?

- Interact with databases via Ecto.Adapters.SQL following the repository pattern.
  - Eg: Postgresql, MySQL, SQLite, Mnesia

- Map data from any source into Elixir Structs, regardless if are backed by a database or not

Database Access

- Repository
  - Single point of source to interact with DB
  - Splits behaviour and data
- Active Record
  - Objects carry both persistent data and behavior which operates on that data.
  - https://guides.rubyonrails.org/active_record_basics.html#the-active-record-pattern

Ecto package Structure

- ecto
  - Data structures
  - Data change tracking

- ecto_sql

  Provides building blocks for writing SQL adapters for Ecto
  - Default implementations for Postgres and MySql
  - Test Sandbox
  - Migrations
  - Dependencies: ecto, telemetry

Main Components

- Repo
- Schema
- Changeset
- Query
- Multi

Ecto.Repo

- Repositories are wrappers around the data store. Via the repository, we can create, update, destroy and query existing entries. A repository needs an adapter and credentials to communicate to the database

Ecto.Schema

- Schemas are used to map any data source into an Elixir struct.

Ecto.Changeset

- Provide a way for developers to filter and cast external parameters, as well as mechanism to track and validate changes before they are applied to your data

Ecto.Query

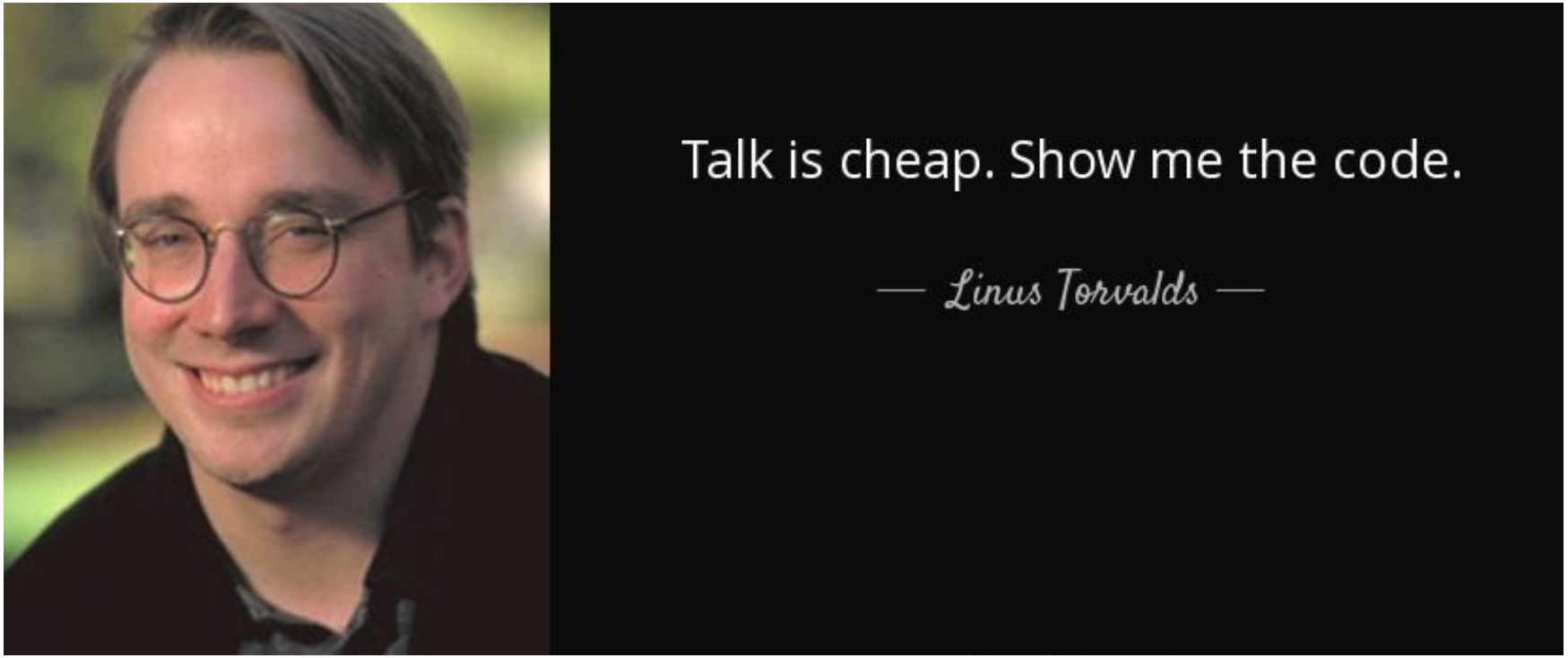
- Written in Elixir syntax, queries are used to retrieve information from a given repository. Queries in Ecto are secure, avoiding common problems like SQL Injection, while still being composable, allowing developers to build queries piece by piece instead of all at once.

Ecto.Multi

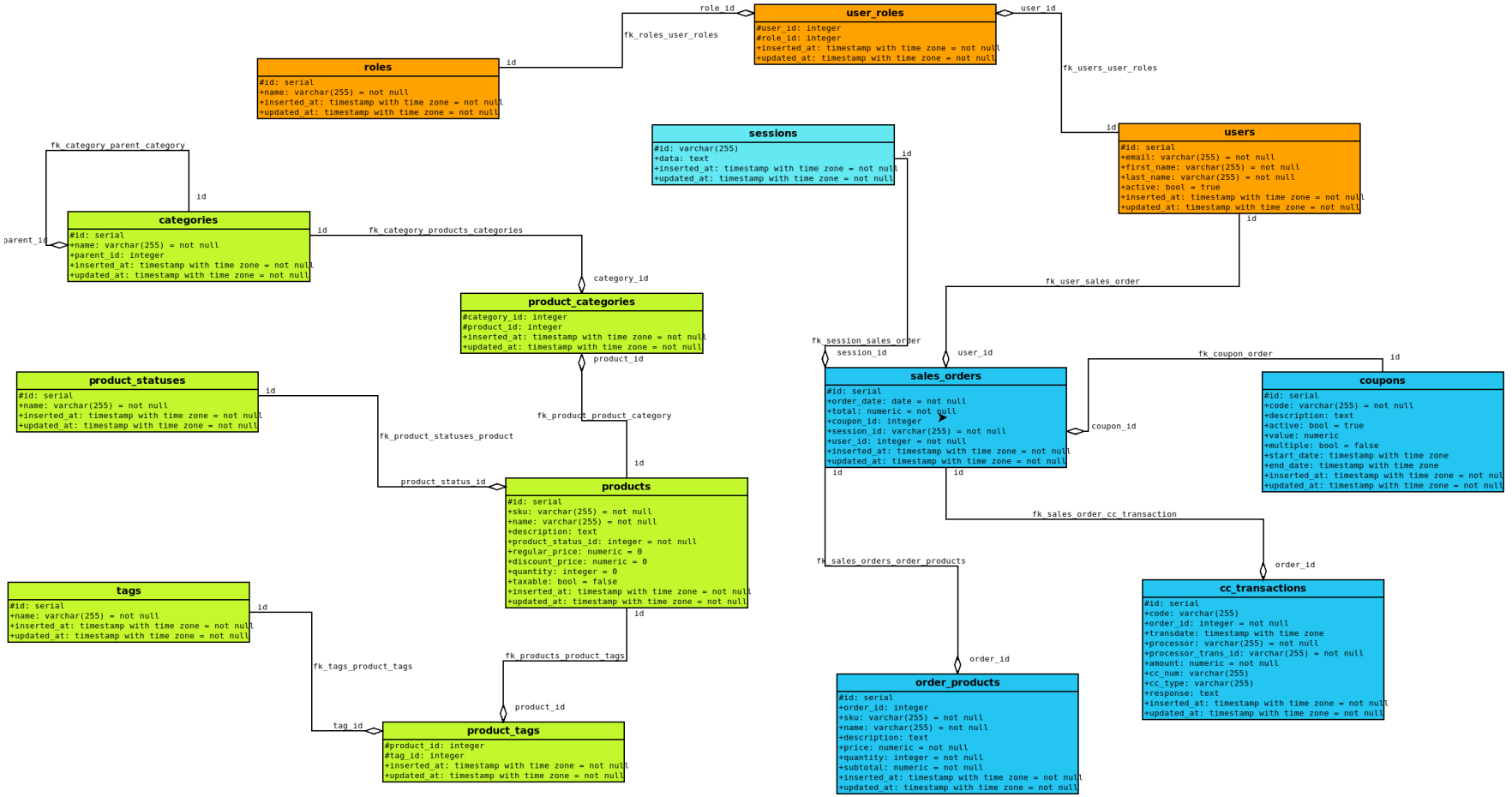
- Used to run transactional operations.
- All or nothing

Testing

- Ecto.Adapters.SQL.Sandbox
  - A pool for concurrent transactional tests
  - Could be configured to run with one connection or shared connections.


Talk is cheap. Show me the code.
— Linus Torvalds —

e-commerce-db



https://github.com/ramortegui/e-commerce-db

```
File  Edit  View  Search  Terminal  Help
~/projects/elixir $ mix new mini_commerce --sup
* creating README.md
* creating .formatter.exs
* creating .gitignore
* creating mix.exs
* creating config
* creating config/config.exs
* creating lib
* creating lib/mini_commerce.ex
* creating lib/mini_commerce/application.ex
* creating test
* creating test/test_helper.exs
* creating test/mini_commerce_test.exs

Your Mix project was created successfully.
You can use "mix" to compile it, test it, and more:

    cd mini_commerce
    mix test

Run "mix help" for more commands.
~/projects/elixir $
```

```
File  Edit  View  Search  Terminal  Help
 24   use Mix.Project
 23
 22   def project do
 21     [
 20       app: :mini_commerce,
 19       version: "0.1.0",
 18       elixir: "~> 1.8",
 17       start_permanent: Mix.env() == :prod,
 16       deps: deps()
 15     ]
 14   end
 13
 12   # Run "mix help compile.app" to learn about applications.
 11   def application do
 10     [
  9       extra_applications: [:logger],
  8       mod: {MiniCommerce.Application, []}
  7     ]
  6   end
  5
  4   # Run "mix help deps" to learn about dependencies.
  3   defp deps do
  2     [
  1       {:ecto_sql, ">= 3.0.4"},
  0       {:postgrex, ">= 0.14.0"}
  1       # {:dep_from_git, git: "https://github.com/elixir-lang/my_dep.git", tag: "0.1.0"}
mix.exs                                                          26,1          25%
-- VISUAL LINE --
```

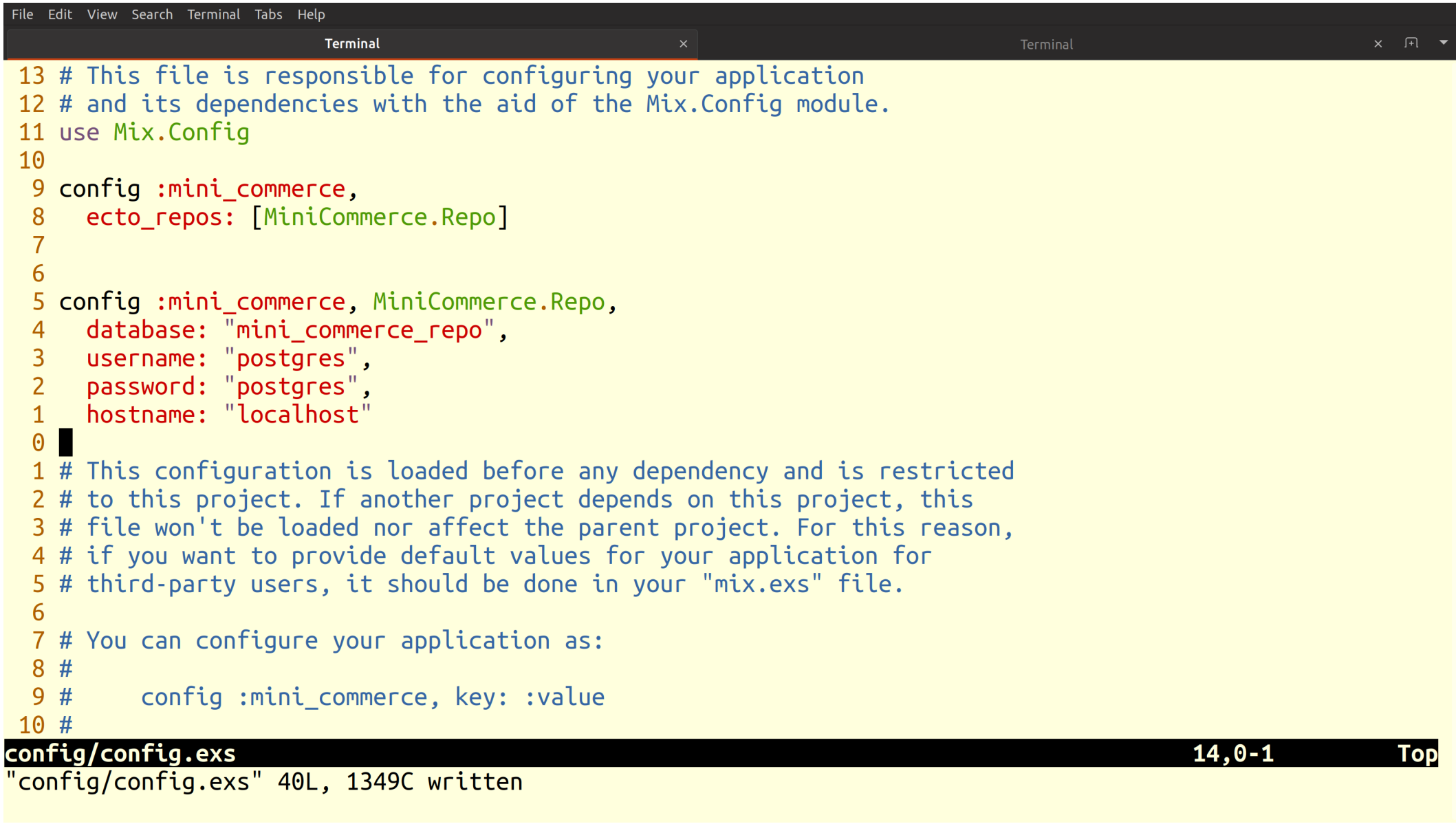
```
File  Edit  View  Search  Terminal  Help
~/projects/elixir/mini_commerce $ mix deps.get
Resolving Hex dependencies...
Dependency resolution completed:
New:
  connection 1.0.4
  db_connection 2.0.3
  decimal 1.6.0
  ecto 3.0.6
  ecto_sql 3.0.4
  postgrex 0.14.1
  telemetry 0.3.0
* Getting ecto_sql (Hex package)
* Getting postgrex (Hex package)
* Getting connection (Hex package)
* Getting db_connection (Hex package)
* Getting decimal (Hex package)
* Getting ecto (Hex package)
* Getting telemetry (Hex package)
~/projects/elixir/mini_commerce $
```

```
File  Edit  View  Search  Terminal  Help
~/projects/elixir/mini_commerce $ git init
Initialized empty Git repository in /home/ramortegui/projects/elixir/mini_commerce/.git/
~/projects/elixir/mini_commerce[master #%] $ git add .
~/projects/elixir/mini_commerce[master +] $ git commit -m "Initial commit"
[master (root-commit) e1988a0] Initial commit
 10 files changed, 165 insertions(+)
 create mode 100644 .formatter.exs
 create mode 100644 .gitignore
 create mode 100644 README.md
 create mode 100644 config/config.exs
 create mode 100644 lib/mini_commerce.ex
 create mode 100644 lib/mini_commerce/application.ex
 create mode 100644 mix.exs
 create mode 100644 mix.lock
 create mode 100644 test/mini_commerce_test.exs
 create mode 100644 test/test_helper.exs
~/projects/elixir/mini_commerce[master] $
```

```
File  Edit  View  Search  Terminal  Help
Compiling 1 file (.ex)
Generated connection app
===> Compiling telemetry
==> decimal
Compiling 1 file (.ex)
Generated decimal app
==> db_connection
Compiling 16 files (.ex)
Generated db_connection app
==> ecto
Compiling 54 files (.ex)
warning: Inspect.Algebra.surround_many/5 is deprecated. Use Inspect.Algebra.container_doc/6 instead
Found at 3 locations:
  lib/ecto/changeset.ex:2755
  lib/ecto/query/inspect.ex:16
  lib/ecto/query/inspect.ex:30

Generated ecto app
==> postgrex
Compiling 61 files (.ex)
Generated postgrex app
==> ecto_sql
Compiling 23 files (.ex)
Generated ecto_sql app
==> mini_commerce
Compiling 2 files (.ex)
Generated mini_commerce app
~/projects/elixir/mini_commerce[master] $
```

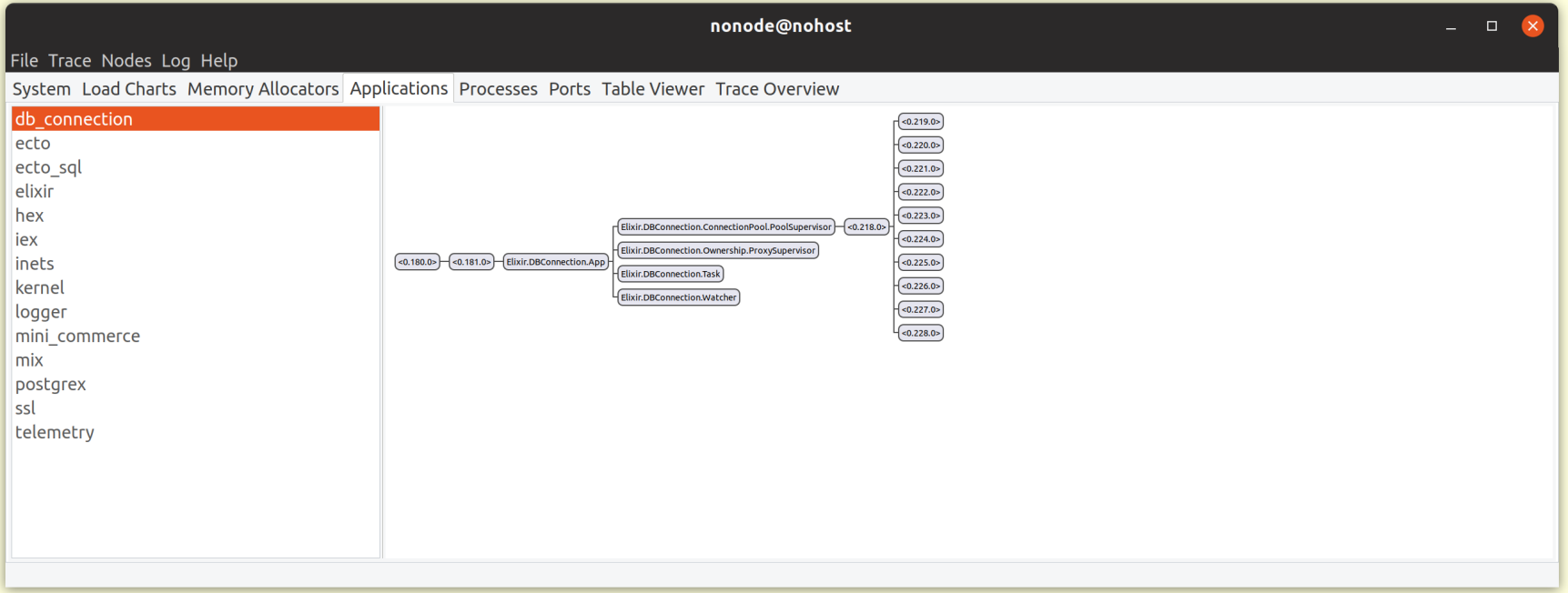
```
File  Edit  View  Search  Terminal  Help
  lib/ecto/query/inspect.ex:30

Generated ecto app
==> postgrex
Compiling 61 files (.ex)
Generated postgrex app
==> ecto_sql
Compiling 23 files (.ex)
Generated ecto_sql app
==> mini_commerce
Compiling 2 files (.ex)
Generated mini_commerce app
~/projects/elixir/mini_commerce[master] $ mix ecto
Ecto v3.0.6
A toolkit for data mapping and language integrated query for Elixir.

Available tasks:

mix ecto.create        # Creates the repository storage
mix ecto.drop          # Drops the repository storage
mix ecto.dump          # Dumps the repository database structure
mix ecto.gen.migration # Generates a new migration for the repo
mix ecto.gen.repo      # Generates a new repository
mix ecto.load          # Loads previously dumped database structure
mix ecto.migrate       # Runs the repository migrations
mix ecto.migrations    # Displays the repository migration status
mix ecto.rollback      # Rolls back the repository migrations
~/projects/elixir/mini_commerce[master] $
```

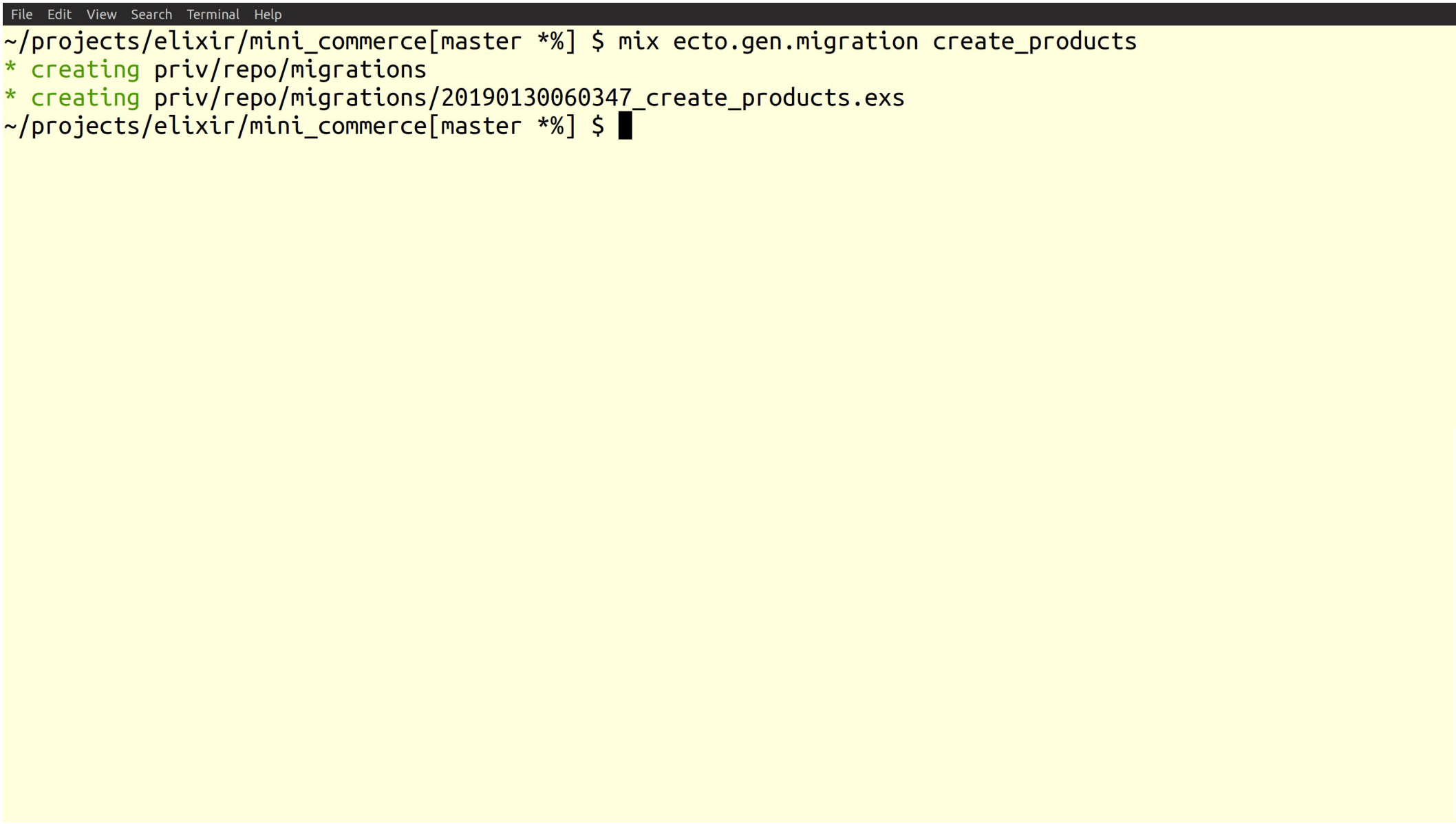
```
File  Edit  View  Search  Terminal  Help
~/projects/elixir/mini_commerce[master] $ mix ecto.gen.repo -r MiniCommerce.Repo
* creating lib/mini_commerce
* creating lib/mini_commerce/repo.ex
* updating config/config.exs
Don't forget to add your new repo to your supervision tree
(typically in lib/mini_commerce/application.ex):

    # For Elixir v1.5 and later
    {MiniCommerce.Repo, []}

    # For Elixir v1.4 and earlier
    supervisor(MiniCommerce.Repo, [])

And to add it to the list of ecto repositories in your
configuration files (so Ecto tasks work as expected):

    config :mini_commerce,
      ecto_repos: [MiniCommerce.Repo]


~/projects/elixir/mini_commerce[master *%] $
```


File Edit View Search Terminal Tabs Help
Terminal
Terminal
13 # This file is responsible for configuring your application
12 # and its dependencies with the aid of the Mix.Config module.
11 use Mix.Config
10
9 config :mini_commerce,
8 ecto_repos: [MiniCommerce.Repo]
7
6
5 config :mini_commerce, MiniCommerce.Repo,
4 database: "mini_commerce_repo",
3 username: "postgres",
2 password: "postgres",
1 hostname: "localhost"
0
1 # This configuration is loaded before any dependency and is restricted
2 # to this project. If another project depends on this project, this
3 # file won't be loaded nor affect the parent project. For this reason,
4 # if you want to provide default values for your application for
5 # third-party users, it should be done in your "mix.exs" file.
6
7 # You can configure your application as:
8 #
9 # config :mini_commerce, key: :value
10 #
config/config.exs 14,0-1 Top
"config/config.exs" 40L, 1349C written

```
File  Edit  View  Search  Terminal  Help
~/projects/elixir/mini_commerce[master *%] $ clear

~/projects/elixir/mini_commerce[master *%] $ mix ecto.create
Compiling 3 files (.ex)
Generated mini_commerce app
The database for MiniCommerce.Repo has been created
~/projects/elixir/mini_commerce[master *%] $ psql -l | grep mini_commerce
 mini_commerce_repo     | postgres   | UTF8     | en_CA.UTF-8 | en_CA.UTF-8 |
~/projects/elixir/mini_commerce[master *%] $
```

```
File  Edit  View  Search  Terminal  Help
~/projects/elixir/mini_commerce[master *%] $ iex -S mix
Erlang/OTP 21 [erts-10.2.3] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [hipe]

Interactive Elixir (1.8.0) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> :observer.start
:ok
iex(2)>
```

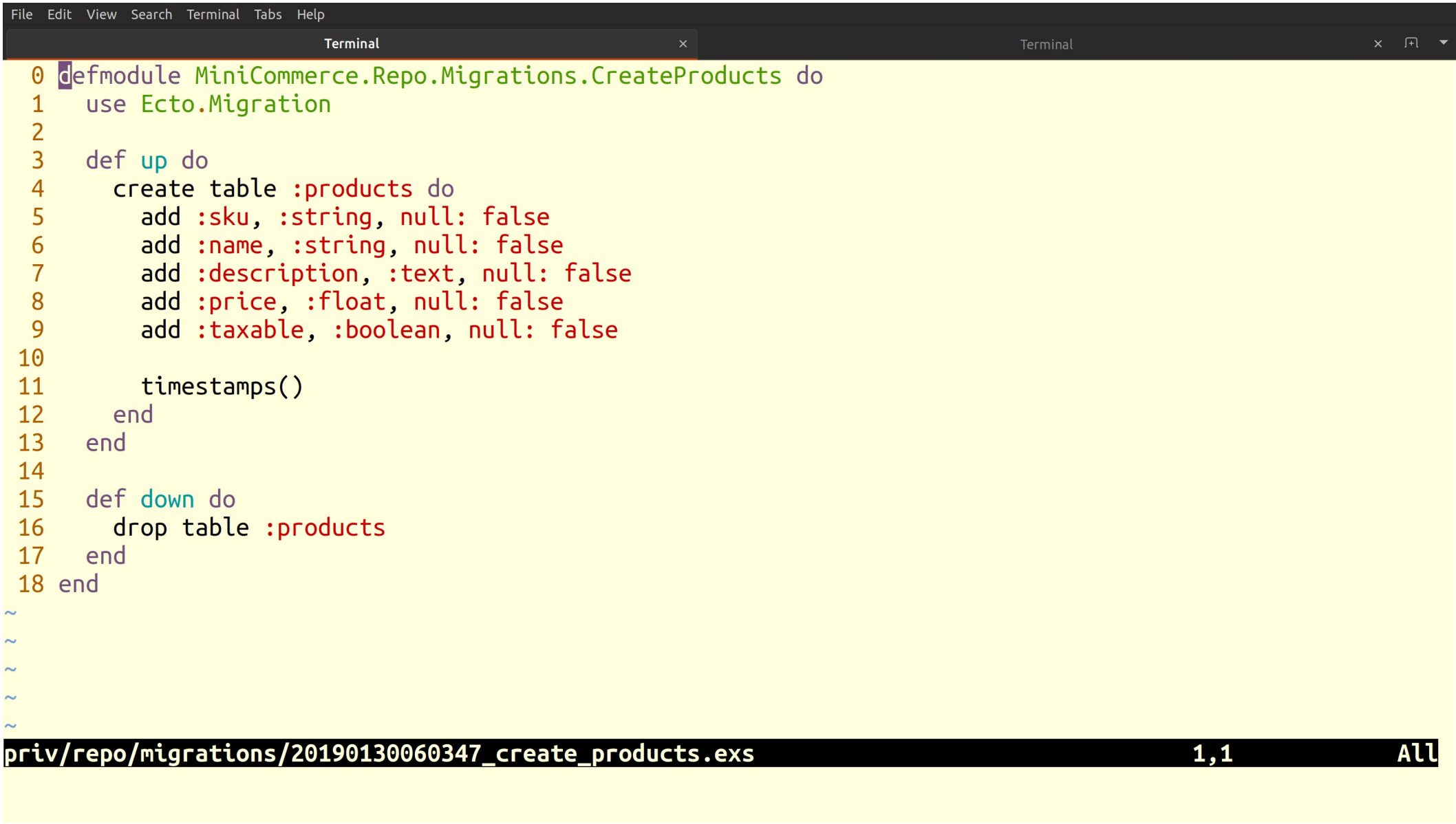


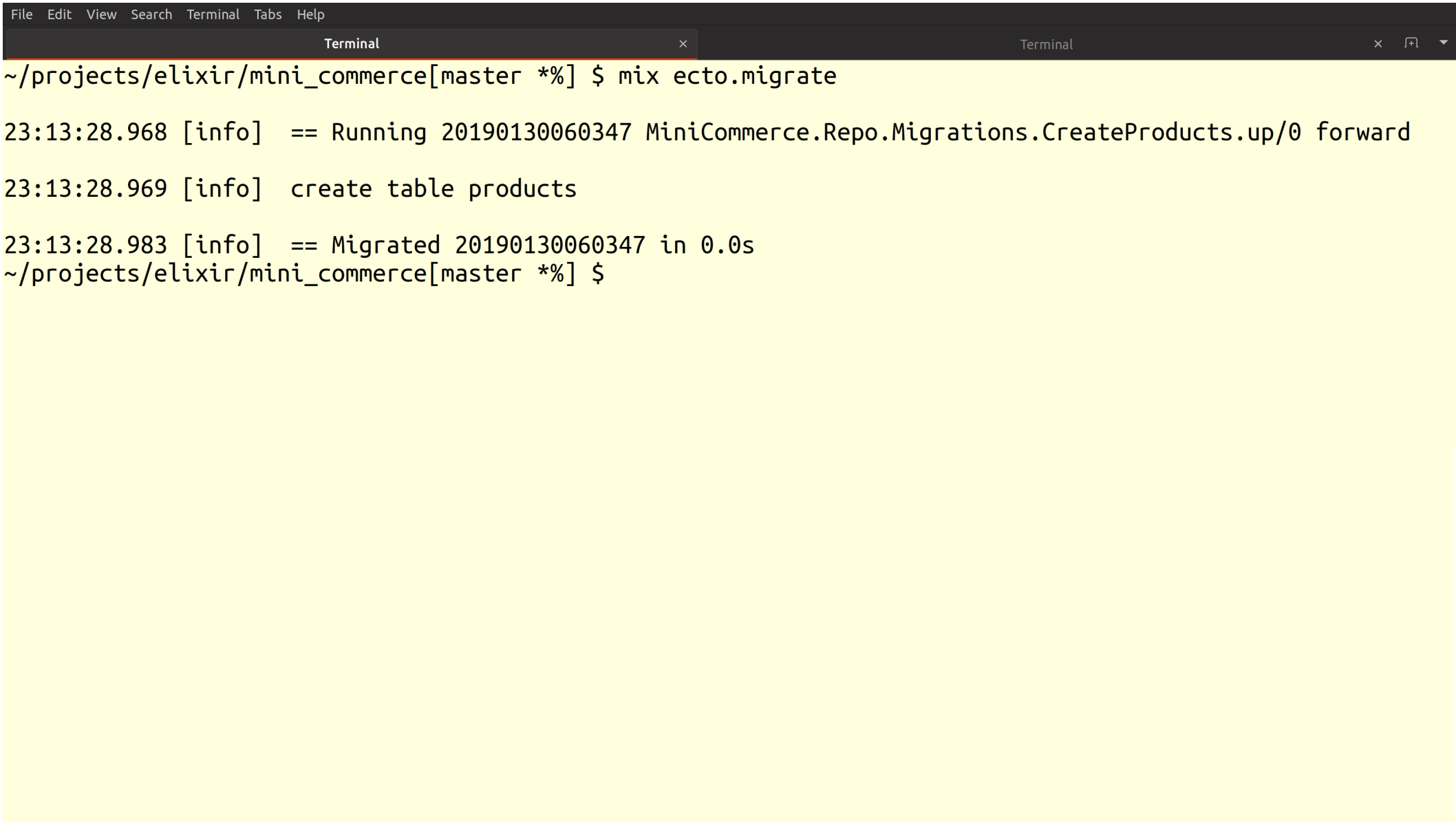

File Edit View Search Terminal Help
~/projects/elixir/mini_commerce[master *%] $ mix ecto.gen.migration create_products
* creating priv/repo/migrations
* creating priv/repo/migrations/20190130060347_create_products.exs
~/projects/elixir/mini_commerce[master *%] $

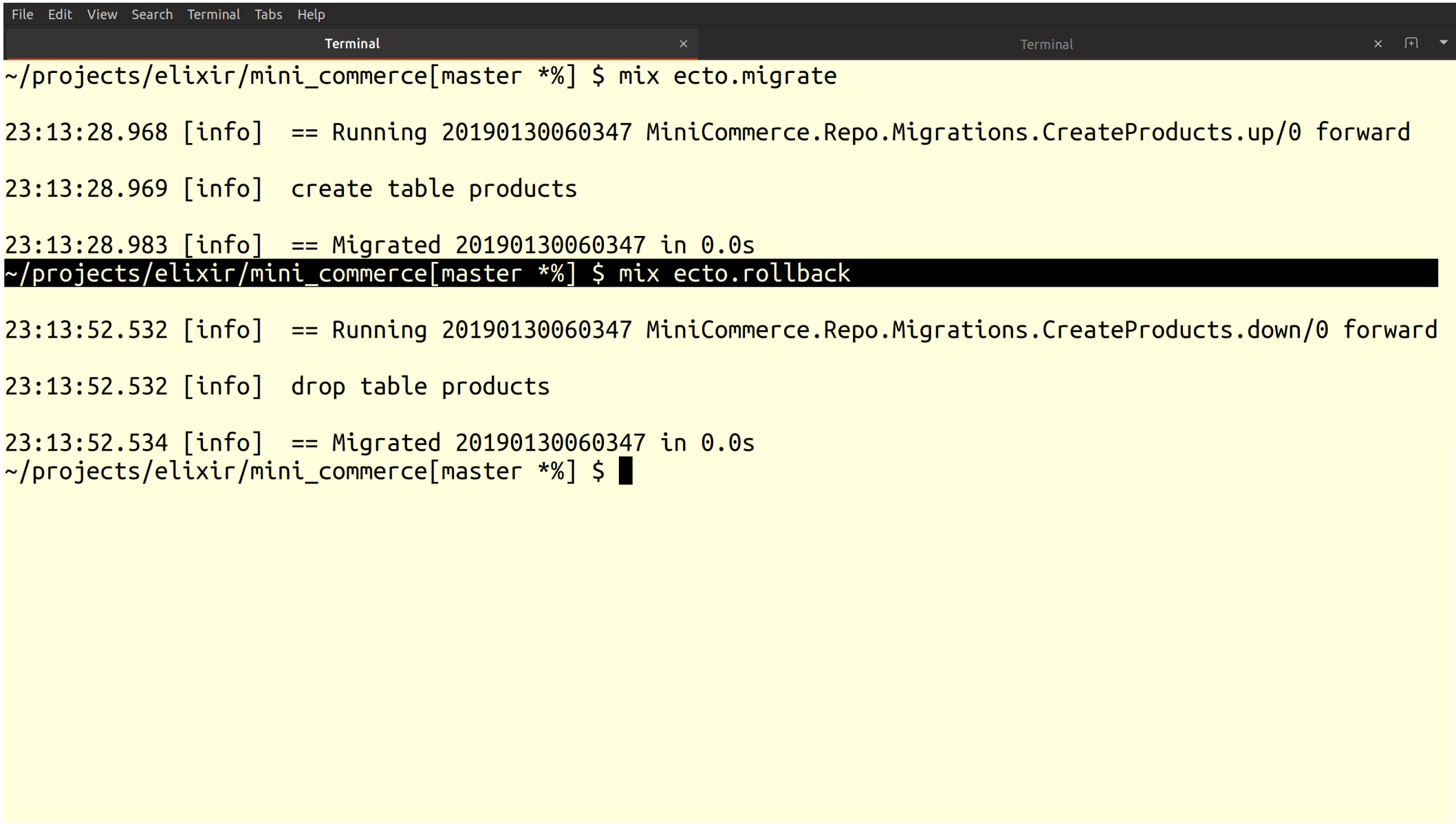
```
File  Edit  View  Search  Terminal  Help
  0 defmodule MiniCommerce.Repo.Migrations.CreateProducts do
  1   use Ecto.Migration
  2
  3   def change do
  4
  5   end
  6 end
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
priv/repo/migrations/20190130060347_create_products.exs                1,1           All
"priv/repo/migrations/20190130060347_create_products.exs" 7L, 106C
```

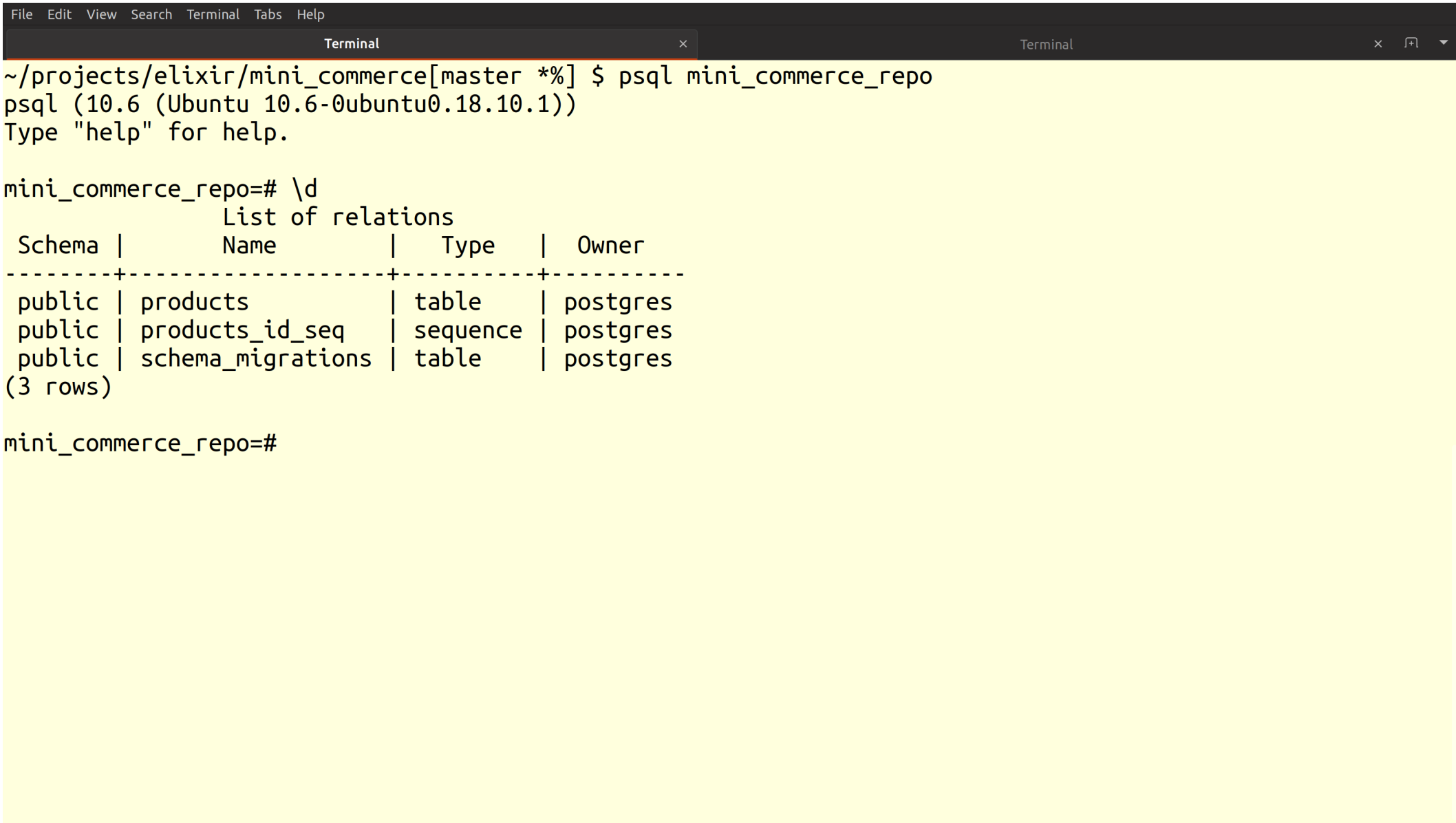
File Edit View Search Terminal Help

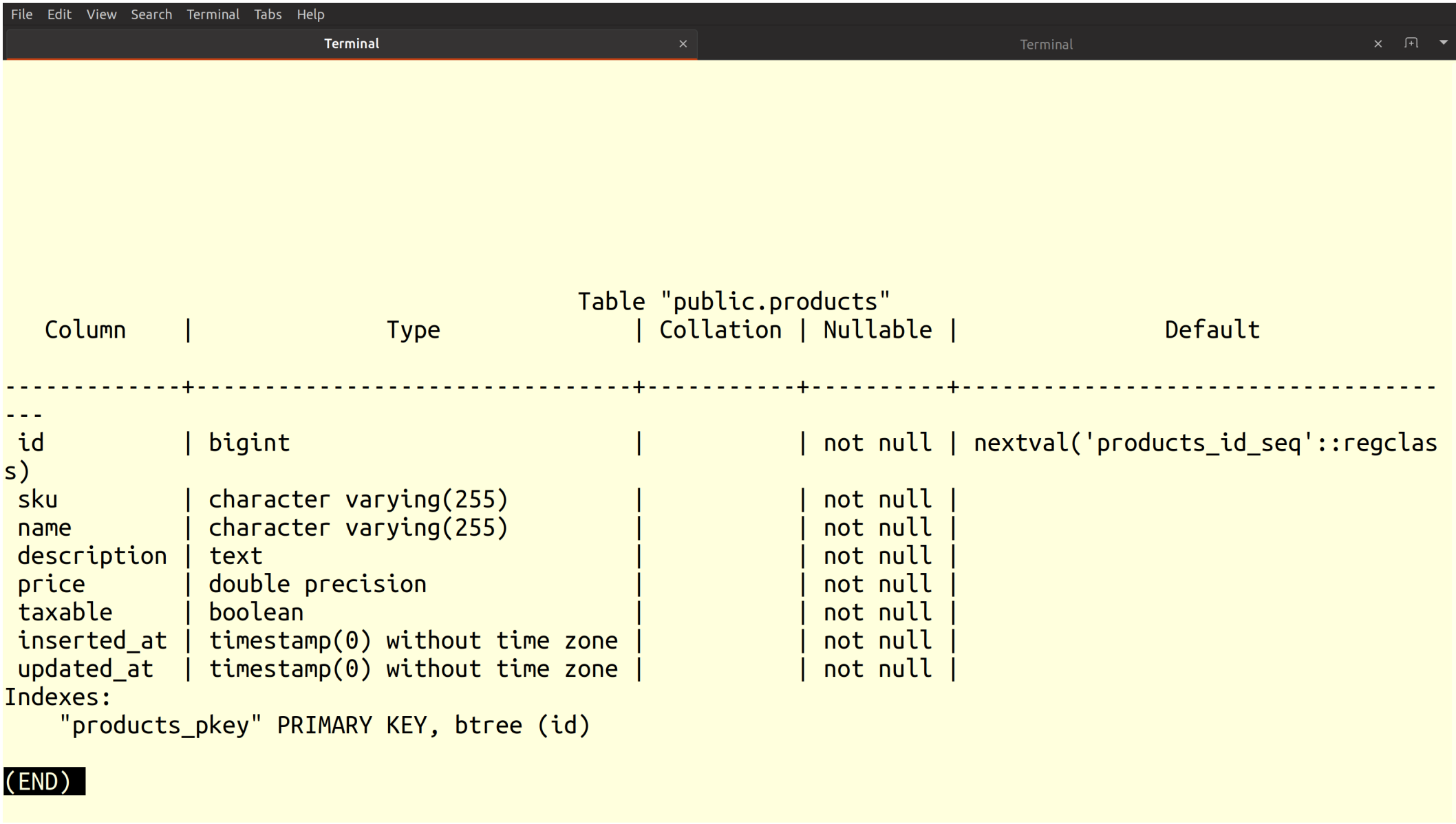
```
 0 defmodule MiniCommerce.Repo.Migrations.CreateProducts do
 1   use Ecto.Migration
 2
 3   def change do
 4     create table :products do
 5       add :sku, :string, null: false
 6       add :name, :string, null: false
 7       add :description, :text, null: false
 8       add :price, :float, null: false
 9       add :taxable, :boolean, null: false
10
11       timestamps()
12     end
13   end
14 end
~
~
~
~
~
~
~
~
~
~
~
~
priv/repo/migrations/20190130060347_create_products.exs                1,1           All
```

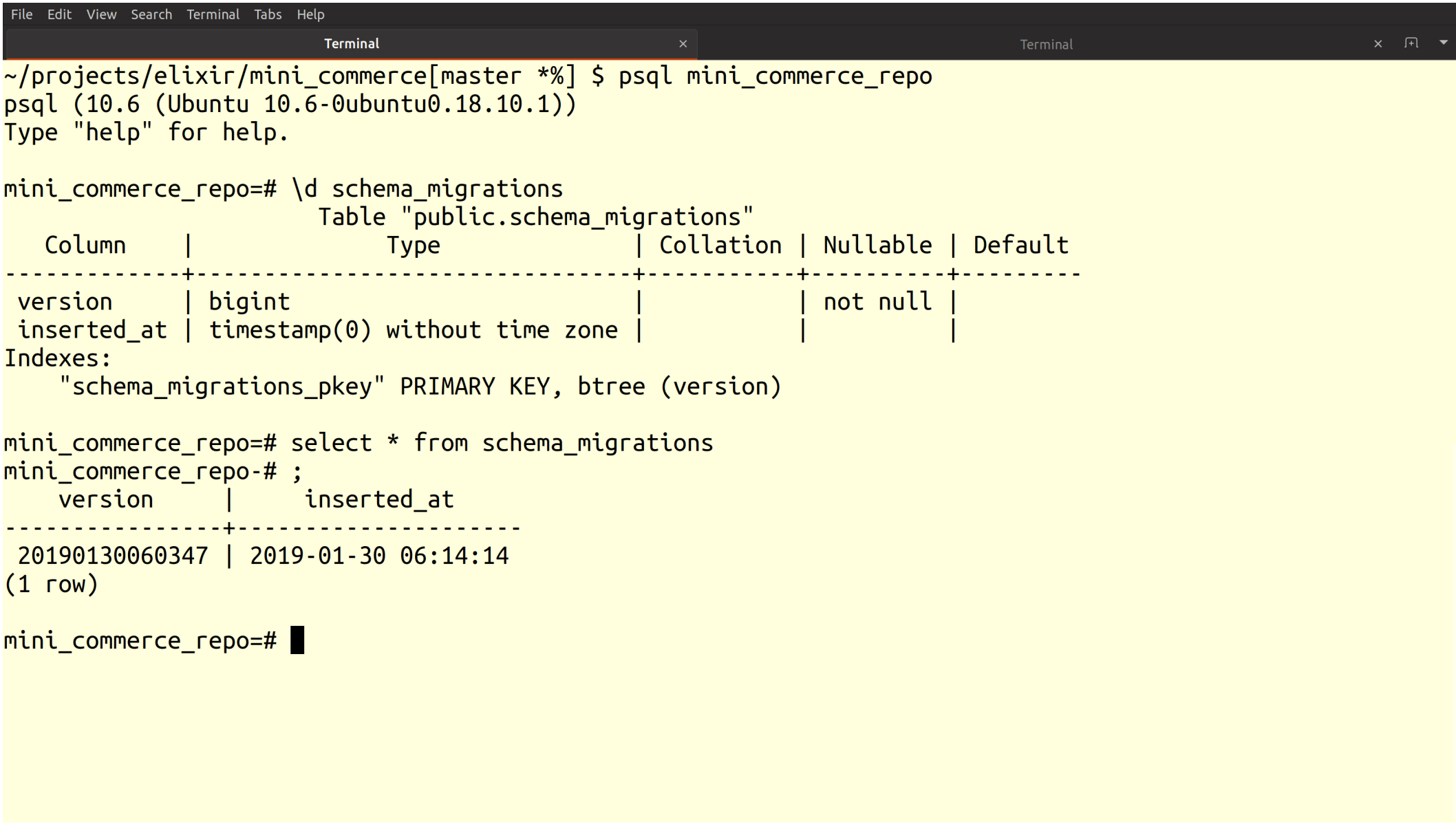

File Edit View Search Terminal Tabs Help
Terminal
Terminal
0 defmodule MiniCommerce.Repo.Migrations.CreateProducts do
1 use Ecto.Migration
2
3 def up do
4 create table :products do
5 add :sku, :string, null: false
6 add :name, :string, null: false
7 add :description, :text, null: false
8 add :price, :float, null: false
9 add :taxable, :boolean, null: false
10
11 timestamps()
12 end
13 end
14
15 def down do
16 drop table :products
17 end
18 end
~
~
~
~
~
priv/repo/migrations/20190130060347_create_products.exs 1,1 All


File Edit View Search Terminal Tabs Help
Terminal
Terminal
~/projects/elixir/mini_commerce[master *%] $ mix ecto.migrate
23:13:28.968 [info] == Running 20190130060347 MiniCommerce.Repo.Migrations.CreateProducts.up/0 forward
23:13:28.969 [info] create table products
23:13:28.983 [info] == Migrated 20190130060347 in 0.0s
~/projects/elixir/mini_commerce[master *%] $


File Edit View Search Terminal Tabs Help
Terminal
Terminal
~/projects/elixir/mini_commerce[master *%] $ mix ecto.migrate
23:13:28.968 [info] == Running 20190130060347 MiniCommerce.Repo.Migrations.CreateProducts.up/0 forward
23:13:28.969 [info] create table products
23:13:28.983 [info] == Migrated 20190130060347 in 0.0s
~/projects/elixir/mini_commerce[master *%] $ mix ecto.rollback
23:13:52.532 [info] == Running 20190130060347 MiniCommerce.Repo.Migrations.CreateProducts.down/0 forward
23:13:52.532 [info] drop table products
23:13:52.534 [info] == Migrated 20190130060347 in 0.0s
~/projects/elixir/mini_commerce[master *%] $


File Edit View Search Terminal Tabs Help
Terminal
Terminal
~/projects/elixir/mini_commerce[master *%] $ psql mini_commerce_repo
psql (10.6 (Ubuntu 10.6-0ubuntu0.18.10.1))
Type "help" for help.
mini_commerce_repo=# \d
List of relations
Schema | Name | Type | Owner
public | products | table | postgres
public | products_id_seq | sequence | postgres
public | schema_migrations | table | postgres
(3 rows)
mini_commerce_repo=#


File Edit View Search Terminal Tabs Help
Terminal
Terminal
Table "public.products"
Column | Type | Collation | Nullable | Default
id | bigint | | not null | nextval('products_id_seq'::regclass)
sku | character varying(255) | | not null |
name | character varying(255) | | not null |
description | text | | not null |
price | double precision | | not null |
taxable | boolean | | not null |
inserted_at | timestamp(0) without time zone | | not null |
updated_at | timestamp(0) without time zone | | not null |
Indexes:
"products_pkey" PRIMARY KEY, btree (id)
(END)


File Edit View Search Terminal Tabs Help
Terminal
Terminal
~/projects/elixir/mini_commerce[master *%] $ psql mini_commerce_repo
psql (10.6 (Ubuntu 10.6-0ubuntu0.18.10.1))
Type "help" for help.
mini_commerce_repo=# \d schema_migrations
Table "public.schema_migrations"
Column | Type | Collation | Nullable | Default
version | bigint | | not null |
inserted_at | timestamp(0) without time zone | | |
Indexes:
"schema_migrations_pkey" PRIMARY KEY, btree (version)
mini_commerce_repo=# select * from schema_migrations
mini_commerce_repo-# ;
version | inserted_at
20190130060347 | 2019-01-30 06:14:14
(1 row)
mini_commerce_repo=#

```
File  Edit  View  Search  Terminal  Help
 14 defmodule MiniCommerce.Product do
 13   use Ecto.Schema
 12
 11   schema "products" do
 10     field :sku, :string
  9     field :name, :string
  8     field :description, :string
  7     field :quantity, :integer
  6     field :price, :float
  5     field :taxable, :boolean
  4
  3     timestamps()
  2   end
  1 end
  0
~
~
~
~
~
~
~
~
~
~
~
~
lib/mini_commerce/products.ex                                  15,0-1         All
"lib/mini_commerce/products.ex" 15L, 270C written
```

```
File  Edit  View  Search  Terminal  Help
iex(1)> %MiniCommerce.Product{}
%MiniCommerce.Product{
  __meta__: #Ecto.Schema.Metadata<:built, "products">,
  description: nil,
  id: nil,
  inserted_at: nil,
  name: nil,
  quantity: nil,
  sku: nil,
  taxable: nil,
  updated_at: nil
}
iex(2)> MiniCommerce.Repo.insert(%MiniCommerce.Product{})

23:37:05.845 [debug] QUERY ERROR db=4.3ms queue=0.5ms
INSERT INTO "products" ("inserted_at","updated_at") VALUES ($1,$2) RETURNING "id" [~N[2019-01-30 06:37:05
], ~N[2019-01-30 06:37:05]]
** (Postgrex.Error) ERROR 23502 (not_null_violation) null value in column "sku" violates not-null constra
int

    table: products
    column: sku

Failing row contains (4, null, null, null, null, 0, null, 2019-01-30 06:37:05, 2019-01-30 06:37:05).
    (ecto_sql) lib/ecto/adapters/sql.ex:620: Ecto.Adapters.SQL.raise_sql_call_error/1
    (ecto) lib/ecto/repo/schema.ex:651: Ecto.Repo.Schema.apply/4
    (ecto) lib/ecto/repo/schema.ex:264: anonymous fn/15 in Ecto.Repo.Schema.do_insert/3
iex(2)>
```

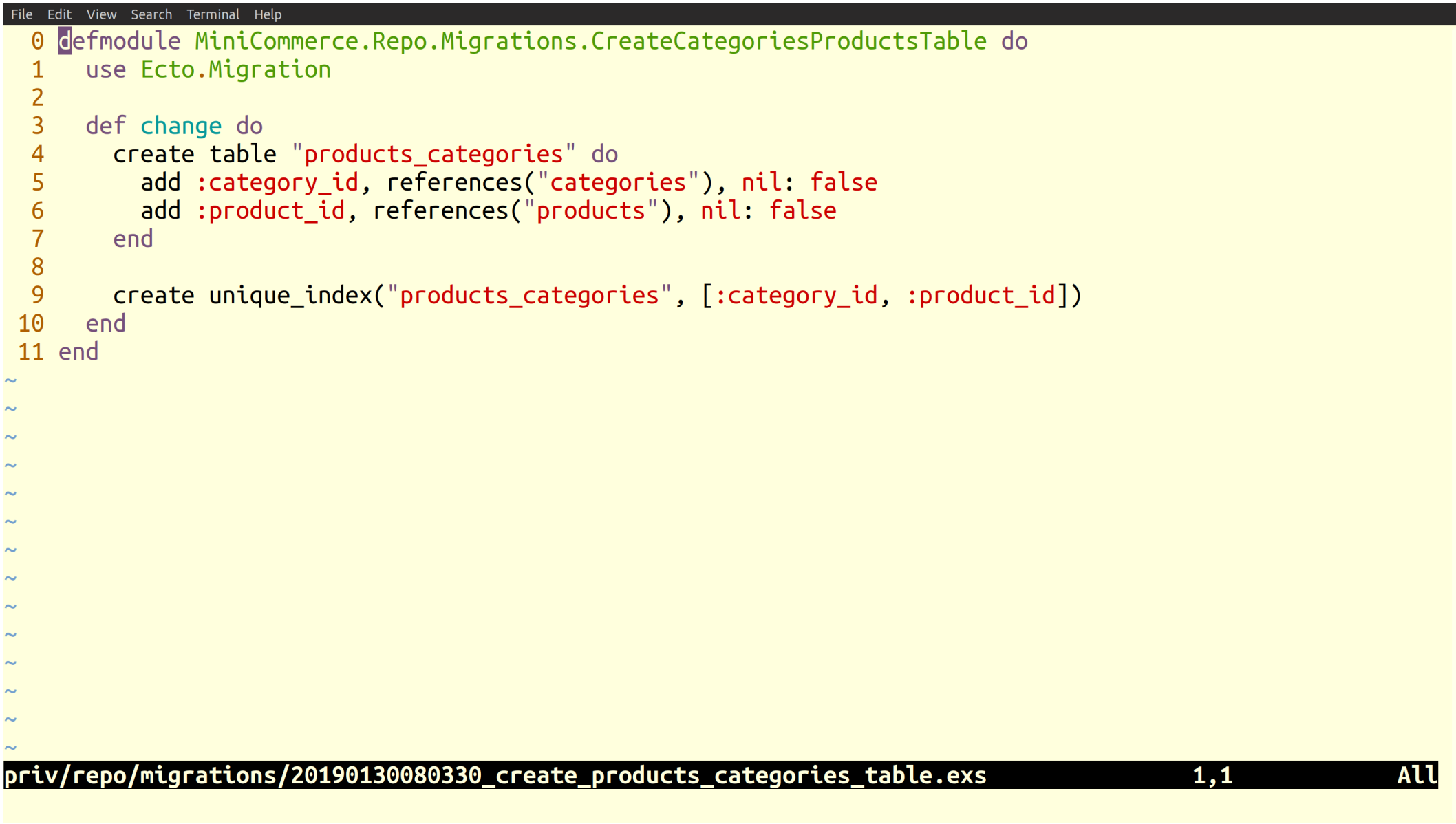
```
File  Edit  View  Search  Terminal  Help
** (Ecto.ChangeError) value `10` for `MiniCommerce.Product.price` in `insert` does not match type :float
    (ecto) lib/ecto/repo/schema.ex:932: Ecto.Repo.Schema.dump_field!/6
    (ecto) lib/ecto/repo/schema.ex:941: anonymous fn/6 in Ecto.Repo.Schema.dump_fields!/5
    (stdlib) maps.erl:257: :maps.fold_1/3
    (ecto) lib/ecto/repo/schema.ex:939: Ecto.Repo.Schema.dump_fields!/5
    (ecto) lib/ecto/repo/schema.ex:872: Ecto.Repo.Schema.dump_changes!/6
    (ecto) lib/ecto/repo/schema.ex:257: anonymous fn/15 in Ecto.Repo.Schema.do_insert/3
iex(1)> MiniCommerce.Repo.insert(%MiniCommerce.Product{sku: "SU01", name: "SoftecUltra", description: "Sa
ndal for beach.", taxable: false, quantity: 10, price: 10.0})

23:40:25.027 [debug] QUERY OK db=10.0ms decode=0.8ms queue=1.1ms
INSERT INTO "products" ("description","name","price","quantity","sku","taxable","inserted_at","updated_at
") VALUES ($1,$2,$3,$4,$5,$6,$7,$8) RETURNING "id" ["Sandal for beach.", "SoftecUltra", 10.0, 10, "SU01",
 false, ~N[2019-01-30 06:40:25], ~N[2019-01-30 06:40:25]]
{:ok,
 %MiniCommerce.Product{
   __meta__: #Ecto.Schema.Metadata<:loaded, "products">,
   description: "Sandal for beach.",
   id: 6,
   inserted_at: ~N[2019-01-30 06:40:25],
   name: "SoftecUltra",
   price: 10.0,
   quantity: 10,
   sku: "SU01",
   taxable: false,
   updated_at: ~N[2019-01-30 06:40:25]
 }}
iex(2)>
```

```
File  Edit  View  Search  Terminal  Help
~/projects/elixir/mini_commerce[master %] $ more .iex.exs
alias MiniCommerce.Repo
alias MiniCommerce.Product
~/projects/elixir/mini_commerce[master %] $ iex -S mix
Erlang/OTP 21 [erts-10.2.3] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [hipe]

Interactive Elixir (1.8.0) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> Repo.all(Product)

23:45:32.341 [debug] QUERY OK source="products" db=5.3ms decode=0.7ms queue=0.6ms
SELECT p0."id", p0."sku", p0."name", p0."description", p0."quantity", p0."price", p0."taxable", p0."inser
ted_at", p0."updated_at" FROM "products" AS p0 []
[
  %MiniCommerce.Product{
    __meta__: #Ecto.Schema.Metadata<:loaded, "products">,
    description: "Sandal for beach.",
    id: 6,
    inserted_at: ~N[2019-01-30 06:40:25],
    name: "SoftecUltra",
    price: 10.0,
    quantity: 10,
    sku: "SU01",
    taxable: false,
    updated_at: ~N[2019-01-30 06:40:25]
  }
]
iex(2)>
```

```
File  Edit  View  Search  Terminal  Help
  0 alias MiniCommerce.Repo
  1 alias MiniCommerce.Product
  2
  3 Repo.insert(
  4   %Product{
  5     sku: "SU01",
  6     name: "SoftecUltra",
  7     description: "Sandal for beach.",
  8     taxable: false,
  9     price: 10.0,
 10     quantity: 10}
 11 )
 12
 13
 14 Repo.insert(
 15   %Product{
 16     sku: "SV01",
 17     name: "SoftecVeteran",
 18     description: "Sandal.",
 19     taxable: true,
 20     price: 10.0,
 21     quantity: 1}
 22 )
~
~
~
priv/repo/seeds.exs                                          1,1           All
"priv/repo/seeds.exs" 23L, 368C
```

File Edit View Search Terminal Help

```
 0 defmodule MiniCommerce.Repo.Migrations.CreateCategories do
 1   use Ecto.Migration
 2
 3   def up do
 4     create table "categories" do
 5       add :name, :string, nil: false
 6       add :category_id, references("categories")
 7
 8       timestamps()
 9     end
10   end
11
12   def down do
13     drop table "categories"
14   end
15 end
~
~
~
~
~
~
~
~
~
~
~
priv/repo/migrations/20190130073912_create_categories.exs                1,1           All
```


File Edit View Search Terminal Help
0 defmodule MiniCommerce.Repo.Migrations.CreateCategoriesProductsTable do
1 use Ecto.Migration
2
3 def change do
4 create table "products_categories" do
5 add :category_id, references("categories"), nil: false
6 add :product_id, references("products"), nil: false
7 end
8
9 create unique_index("products_categories", [:category_id, :product_id])
10 end
11 end
priv/repo/migrations/20190130080330_create_products_categories_table.exs 1,1 All

```
File  Edit  View  Search  Terminal  Help
 0 defmodule MiniCommerce.Product do
 1   use Ecto.Schema
 2   import Ecto.Changeset
 3
 4   alias MiniCommerce.Category
 5
 6   schema "products" do
 7     field :sku, :string
 8     field :name, :string
 9     field :description, :string
10     field :quantity, :integer
11     field :price, :float
12     field :taxable, :boolean
13
14     many_to_many :categories, Category, join_through: "products_categories"
15
16     timestamps()
17   end
18
19   def changeset(product, params) do
20     product
21     |> cast(params, [:sku, :name, :description, :quantity, :price, :taxable])
22     |> validate_required([:sku, :name, :description, :quantity, :price, :taxable])
23     |> unique_constraint(:sku)
24     |> validate_number(:quantity, greater_than: 0)
25     |> validate_number(:price, greater_than: 0)
~/projects/elixir/mini_commerce/lib/mini_commerce/product.ex                1,33          Top
"~/projects/elixir/mini_commerce/lib/mini_commerce/product.ex" 28L, 747C written
```

```
File  Edit  View  Search  Terminal  Help

iex(16)> MiniCommerce.Category  |> Repo.all

01:34:12.278 [debug] QUERY OK source="categories" db=2.1ms
SELECT c0."id", c0."name", c0."category_id", c0."inserted_at", c0."updated_at" FROM "categories" AS c0 []
[
  %MiniCommerce.Category{
    __meta__: #Ecto.Schema.Metadata<:loaded, "categories">,
    category: #Ecto.Association.NotLoaded<association :category is not loaded>,
    category_id: nil,
    id: 1,
    inserted_at: ~N[2019-01-30 08:18:36],
    name: "Clothes",
    products: #Ecto.Association.NotLoaded<association :products is not loaded>,
    updated_at: ~N[2019-01-30 08:18:36]
  },
  %MiniCommerce.Category{
    __meta__: #Ecto.Schema.Metadata<:loaded, "categories">,
    category: #Ecto.Association.NotLoaded<association :category is not loaded>,
    category_id: 1,
    id: 2,
    inserted_at: ~N[2019-01-30 08:18:36],
    name: "Footwear",
    products: #Ecto.Association.NotLoaded<association :products is not loaded>,
    updated_at: ~N[2019-01-30 08:18:36]
  }
]
iex(17)>
```

```
File  Edit  View  Search  Terminal  Help
    category_id: nil,
    id: 1,
    inserted_at: ~N[2019-01-30 08:18:36],
    name: "Clothes",
    products: #Ecto.Association.NotLoaded<association :products is not loaded>,
    updated_at: ~N[2019-01-30 08:18:36]
  },
  %MiniCommerce.Category{
    __meta__: #Ecto.Schema.Metadata<:loaded, "categories">,
    category: %MiniCommerce.Category{
      __meta__: #Ecto.Schema.Metadata<:loaded, "categories">,
      category: #Ecto.Association.NotLoaded<association :category is not loaded>,
      category_id: nil,
      id: 1,
      inserted_at: ~N[2019-01-30 08:18:36],
      name: "Clothes",
      products: #Ecto.Association.NotLoaded<association :products is not loaded>,
      updated_at: ~N[2019-01-30 08:18:36]
    },
    category_id: 1,
    id: 2,
    inserted_at: ~N[2019-01-30 08:18:36],
    name: "Footwear",
    products: #Ecto.Association.NotLoaded<association :products is not loaded>,
    updated_at: ~N[2019-01-30 08:18:36]
  }
]
iex(8)> MiniCommerce.Category  |> Repo.all |> Repo.preload(:category)
```

```
File  Edit  View  Search  Terminal  Help
iex(21)> query = from p in Product, where: p.id > 1
#Ecto.Query<from p0 in MiniCommerce.Product, where: p0.id > 1>
iex(22)> Repo.all
all/1    all/2
iex(22)> Repo.all(query)

01:37:47.089 [debug] QUERY OK source="products" db=0.7ms queue=0.8ms
SELECT p0."id", p0."sku", p0."name", p0."description", p0."quantity", p0."price", p0."taxable", p0."inser
ted_at", p0."updated_at" FROM "products" AS p0 WHERE (p0."id" > 1) []
[
  %MiniCommerce.Product{
    __meta__: #Ecto.Schema.Metadata<:loaded, "products">,
    categories: #Ecto.Association.NotLoaded<association :categories is not loaded>,
    description: "Sandal.",
    id: 2,
    inserted_at: ~N[2019-01-30 08:18:36],
    name: "SoftecVeteran",
    price: 10.0,
    quantity: 1,
    sku: "SV01",
    taxable: true,
    updated_at: ~N[2019-01-30 08:18:36]
  }
]
iex(23)>
```

```
File  Edit  View  Search  Terminal  Help
iex(46)> query = from p in "products", where: p.id > 1, select: [:name]
#Ecto.Query<from p0 in "products", where: p0.id > 1, select: [:name]>
iex(47)> Repo.all(query)

01:41:25.737 [debug] QUERY OK source="products" db=0.8ms
SELECT p0."name" FROM "products" AS p0 WHERE (p0."id" > 1) []
[%{name: "SoftecVeteran"}]
iex(48)>
```

```
File  Edit  View  Search  Terminal  Help
iex(49)> query = from p in Product, where: p.id > 1, select: [:name]
#Ecto.Query<from p0 in MiniCommerce.Product, where: p0.id > 1, select: [:name]>
iex(50)> Repo.all(query)

01:42:13.387 [debug] QUERY OK source="products" db=1.5ms
SELECT p0."name" FROM "products" AS p0 WHERE (p0."id" > 1) []
[
  %MiniCommerce.Product{
    __meta__: #Ecto.Schema.Metadata<:loaded, "products">,
    categories: #Ecto.Association.NotLoaded<association :categories is not loaded>,
    description: nil,
    id: nil,
    inserted_at: nil,
    name: "SoftecVeteran",
    price: nil,
    quantity: nil,
    sku: nil,
    taxable: nil,
    updated_at: nil
  }
]
iex(51)>
```

```
File  Edit  View  Search  Terminal  Help
#Ecto.Query<from p0 in MiniCommerce.Product, where: p0.id > 1, select: [:name]>
iex(50)> Repo.all(query)

01:42:13.387 [debug] QUERY OK source="products" db=1.5ms
SELECT p0."name" FROM "products" AS p0 WHERE (p0."id" > 1) []
[
  %MiniCommerce.Product{
    __meta__: #Ecto.Schema.Metadata<:loaded, "products">,
    categories: #Ecto.Association.NotLoaded<association :categories is not loaded>,
    description: nil,
    id: nil,
    inserted_at: nil,
    name: "SoftecVeteran",
    price: nil,
    quantity: nil,
    sku: nil,
    taxable: nil,
    updated_at: nil
  }
]
iex(51)> query = from p in "products", where: p.id > 1, select: [count(p.id)]
#Ecto.Query<from p0 in "products", where: p0.id > 1, select: [count(p0.id)]>
iex(52)> Repo.all(query)

01:42:53.576 [debug] QUERY OK source="products" db=1.0ms queue=1.4ms
SELECT count(p0."id") FROM "products" AS p0 WHERE (p0."id" > 1) []
[[1]]
iex(53)>
```

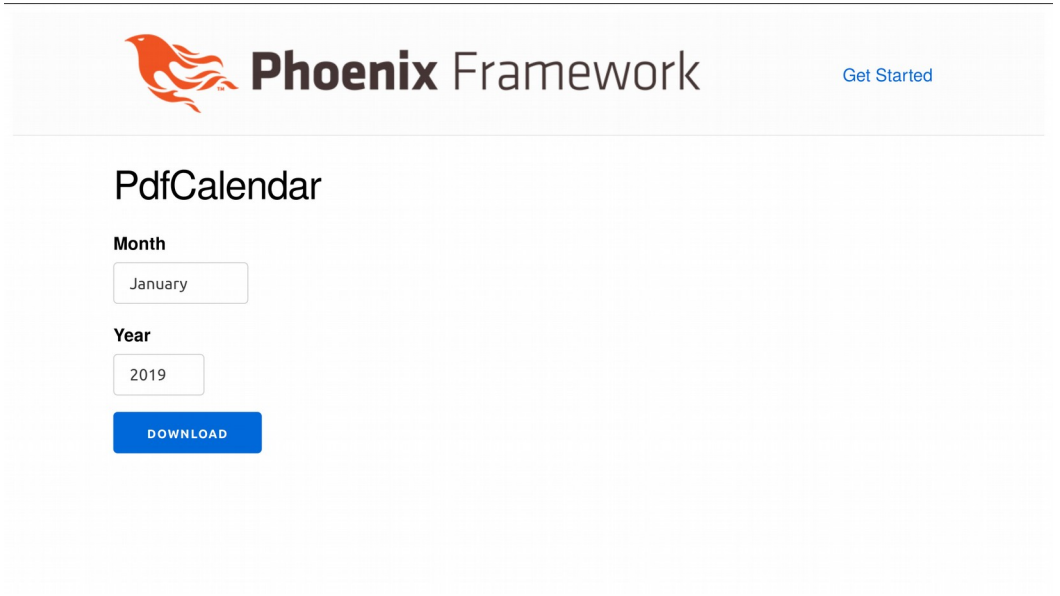
```
File  Edit  View  Search  Terminal  Help
iex(64)> query = from p in "products", where: p.id > 1, select: [count(p.id)]
#Ecto.Query<from p0 in "products", where: p0.id > 1, select: [count(p0.id)]>
iex(65)> Repo.all(query)

01:43:15.066 [debug] QUERY OK source="products" db=1.8ms
SELECT count(p0."id") FROM "products" AS p0 WHERE (p0."id" > 1) []
[[1]]
iex(66)>
```

```
File  Edit  View  Search  Terminal  Help
Erlang/OTP 21 [erts-10.2.3] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [h
ipe]

Interactive Elixir (1.8.1) - press Ctrl+C to exit (type h() ENTER for help)
iex(1)> product = Repo.get!(Product, 1)

16:29:07.803 [debug] QUERY OK source="products" db=4.3ms decode=0.5ms queue=0.6ms
SELECT p0."id", p0."sku", p0."name", p0."description", p0."quantity", p0."price", p0."ta
xable", p0."inserted_at", p0."updated_at" FROM "products" AS p0 WHERE (p0."id" = $1) [1]
%MiniCommerce.Product{
  __meta__: #Ecto.Schema.Metadata<:loaded, "products">,
  categories: #Ecto.Association.NotLoaded<association :categories is not loaded>,
  description: "Sandal for beach.",
  id: 1,
  inserted_at: ~N[2019-01-30 23:27:27],
  name: "SoftecUltra",
  price: 10.0,
  quantity: 10,
  sku: "SU01",
  taxable: false,
  updated_at: ~N[2019-01-30 23:27:27]
}
iex(2)>
```

```
File  Edit  View  Search  Terminal  Help
iex(2)> product_changeset = Product.changeset(product, %{name: "new name"})
#Ecto.Changeset<
  action: nil,
  changes: %{name: "new name"},
  errors: [],
  data: #MiniCommerce.Product<>,
  valid?: true
>
iex(3)> Repo.update(product_changeset)

16:29:36.362 [debug] QUERY OK db=14.6ms queue=0.6ms
UPDATE "products" SET "name" = $1, "updated_at" = $2 WHERE "id" = $3 ["new name", ~N[2019-01-30 23:29:36]
, 1]
{:ok,
 %MiniCommerce.Product{
   __meta__: #Ecto.Schema.Metadata<:loaded, "products">,
   categories: #Ecto.Association.NotLoaded<association :categories is not loaded>,
   description: "Sandal for beach.",
   id: 1,
   inserted_at: ~N[2019-01-30 23:27:27],
   name: "new name",
   price: 10.0,
   quantity: 10,
   sku: "SU01",
   taxable: false,
   updated_at: ~N[2019-01-30 23:29:36]
 }}
iex(4)>
```

Ecto without DB

- Validate Data
- PdfCalendar

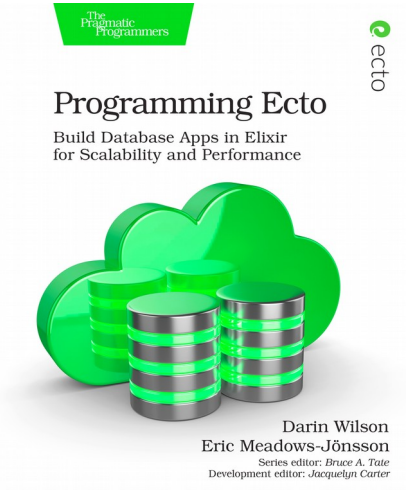


```
File  Edit  View  Search  Terminal  Help
 0 defmodule PdfCalendarWeb.PageController do
 1   use PdfCalendarWeb, :controller
 2
 3   def index(conn, _params) do
 4     changeset = PdfCalendar.Pcal.changeset(pdf_calendar_struct())
 5     render(conn, "index.html", changeset: changeset)
 6   end
 7
 8   def download(conn, params) do
 9     changeset = PdfCalendar.Pcal.changeset(%PdfCalendar.Pcal{}, params["pcal"])
10
11     case changeset.valid? do
12       true ->
13         month = changeset.params["month"]
14         year = changeset.params["year"]
15         {:ok, output} = Pcal.generate_pdf(%Pcal{month: month, year: year, output: pdf_name()})
16         send_download(conn, {:file, output}, filename: "#{month}_#{year}.pdf")
17
18       false ->
19         render(conn, "index.html", changeset: %{changeset | action: 'create'})
20     end
21   end
22
23   defp pdf_calendar_struct do
24     date = DateTime.utc_now()
25
26     %PdfCalendar.Pcal{
27       month: date.month,
28       year: date.year
29     }
30   end
lib/pdf_calendar_web/controllers/page_controller.ex                1,1           Top
```

Summary

- No surprices, all should be explicit.
- OTP application to connect with databases
- The user can map data and give access explicitly
- Give the opportunity to run tests with shared or independent connections
- Provide tools to evolve a database keeping tracking of the changes

Links



- ElixirConf 2017

  Darin Wilson: Thinking In Ecto

  https://www.youtube.com/watch?v=YQxopjai0CU
- Code Beam STO 2018

  Eric Meadows Jöhnson: Ecto - database library for Elixir

  https://www.youtube.com/watch?v=RT4p_g0SLUU
- https://hexdocs.pm/ecto/Ecto.html
- https://github.com/elixir-ecto/ecto

References

- https://hexdocs.pm/ecto/Ecto.html
- https://github.com/elixir-ecto/ecto
- Programming Ecto. Daring Wilson, Eric Meadows Jöhnson. B6. Jan 2019

Thanks!

Q & A?

@ramortegui

https://www.rubenamortegui.com

https://github.com/ramortegui