Interactive - 09 - add a table with state codes

A check constraint on all the state codes would be very long. There are also times when that is just not the best way to check that a value is correct.

There is a different way to check that a value is in a range of values. That is to add a new table that contains all the values and have the database verify that new values are in this list.

First let's create the table with the values. We can also have the database verify that nobody puts in an empty state name. This is the "NOT NULL" on each of the types.

```
CREATE TABLE us_state (
    state_id UUID NOT NULL DEFAULT uuid_generate_v4() primary key,
    state_name text NOT NULL,
    state char varying (2) NOT NULL,
    FIPS_code char varying (2) NOT NULL
);
```

We will also need an index on the state code so that we can reference it.

```
CREATE UNIQUE INDEX us_state_uidx1 on us_state ( state );
```

If this is not the first time this is run you may need to delete existing data.

```
DELETE FROM us_state ;
```

Then we need to populate the state table with states. Please cut-and-paste this or use the supplied file to insert the states.

```
INSERT INTO us_state ( state_name, state, FIPS_code ) values
    ('Alabama','AL','01'),
    ('Alaska','AK','02'),
    ('Arizona','AZ','04'),
    ('Arkansas','AR','05'),
    ('California','CA','06'),
    ('Colorado','CO','08'),
    ('Connecticut','CT','09'),
    ('Delaware','DE','10'),
    ('District of Columbia','DC','11'),
    ('Florida','FL','12'),
    ('Georgia','GA','13'),
    ('Hawaii','HI','15'),
    ('Idaho','ID','16'),
    ('Illinois','IL','17'),
    ('Indiana','IN','18'),
    ('Iowa','IA','19'),
    ('Kansas','KS','20'),
    ('Kentucky','KY','21'),
    ('Louisiana','LA','22'),
    ('Maine','ME','23'),
    ('Maryland','MD','24'),
    ('Massachusetts','MA','25'),
    ('Michigan','MI','26'),
    ('Minnesota','MN','27'),
    ('Mississippi','MS','28'),
    ('Missouri','MO','29'),
    ('Montana','MT','30'),
    ('Nebraska','NE','31'),
    ('Nevada','NV','32'),
    ('New Hampshire','NH','33'),
    ('New Jersey','NJ','34'),
    ('New Mexico','NM','35'),
    ('New York','NY','36'),
    ('North Carolina','NC','37'),
    ('North Dakota','ND','38'),
    ('Ohio','OH','39'),
    ('Oklahoma','OK','40'),
    ('Oregon','OR','41'),
    ('Pennsylvania','PA','42'),
    ('Puerto Rico','PR','72'),
    ('Rhode Island','RI','44'),
    ('South Carolina','SC','45'),
    ('South Dakota','SD','46'),
    ('Tennessee','TN','47'),
    ('Texas','TX','48'),
    ('Utah','UT','49'),
    ('Vermont','VT','50'),
    ('Virginia','VA','51'),
    ('Virgin Islands','VI','78'),
    ('Washington','WA','53'),
    ('West Virginia','WV','54'),
    ('Wisconsin','WI','55'),
    ('Wyoming','WY','56')
;
```

Now let's make the relationship between them know to the database. This is called a foreign key constraint.

```
ALTER TABLE name_list
    ADD CONSTRAINT name_list_state_fk
    FOREIGN KEY (state)
    REFERENCES us_state (state)
;
```

Now when we try to insert data that is has an invalid state code into our list of names it will get rejected.

The following SQL will *not* run successfully

```
insert into name_list ( real_name, state ) values
    ( 'nobody', 'XX' )
;
```

You should get an error that looks similar to:

```
ERROR:  insert or update on table "name_list" violates foreign key constraint "name_list_sta
DETAIL:  Key (state)=(XX) is not present in table "us_state".
```

Book: page 107, 136

FilesToRun: hw09_1.sql

FilesToRun: hw09_2.sql

FilesToRun: hw09_3.sql

FilesToRun: hw09_4.sql