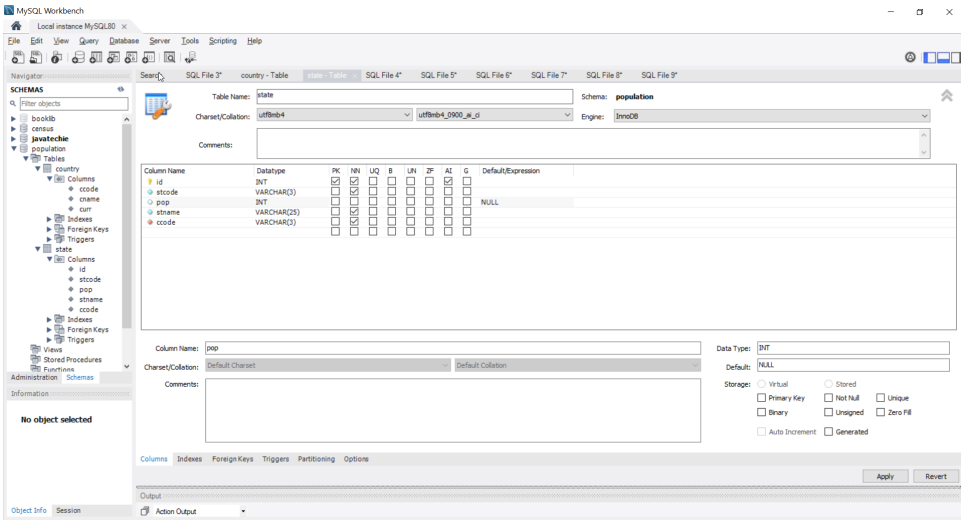
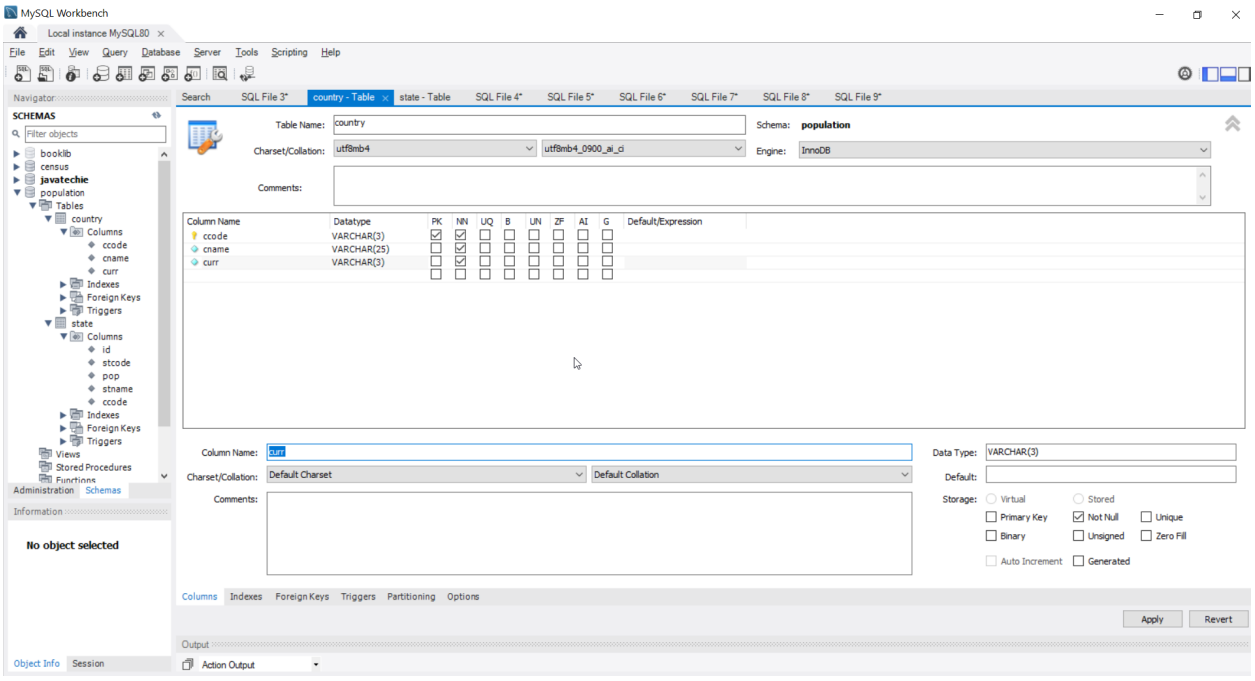
Database -

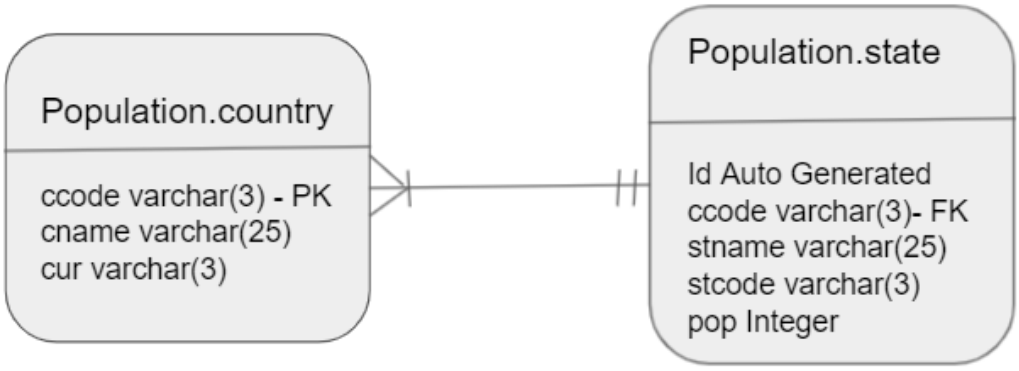
Create a database of your desired flavor that can store simple data based on the following fictional country and state information:

Mysql is used for Database schema -





A below schema is created -



One-to-many relationship

Data.sql and schema.sql are

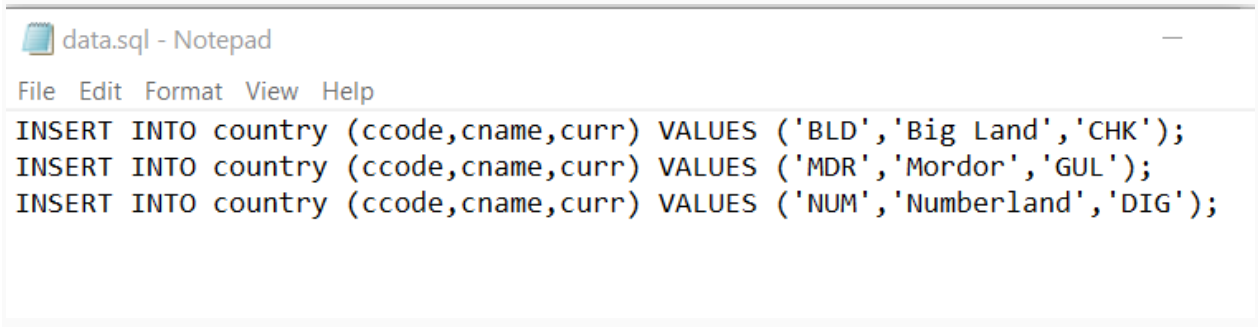
`spring.jpa.hibernate.ddl-auto=none`

This property need to be set in application.properties file to allow spring to create the scheme and not create the spring jpa by itself.

schema.sql - Notepad

File Edit Format View Help

```
CREATE TABLE population.country (
    ccode varchar(3)  NOT NULL ,
    cname VARCHAR(25) NOT NULL,
    PRIMARY KEY (ccode)
);

CREATE TABLE population.state (
    id   INTEGER      NOT NULL AUTO_INCREMENT,
    stcode varchar(3) NOT NULL,
    pop INTEGER,
    stname VARCHAR(25) NOT NULL,
    ccode varchar(3)  NOT NULL,
    FOREIGN KEY (CCODE)

);
```



Similarly the state table is also loaded

The @sql can be used to load the data instead of manually creating -

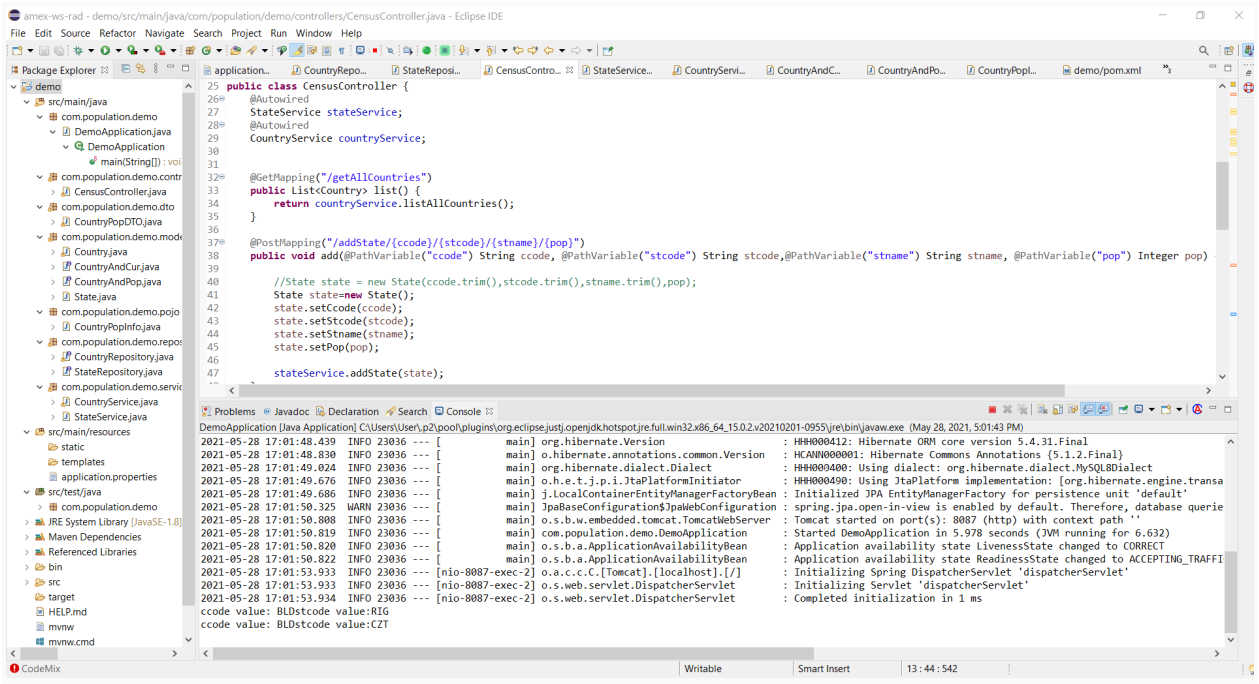
```
@Sql({"/population_schema.sql", "/import_data.sql"})
public class SpringBootInitialLoadIntegrationTest {

    @Autowired
    private CountryRepository countryRepository;

    @Test
    public void testLoadDataForTestClass() {
        assertEquals(3, countryRepository.findAll().size());
    }
}
```

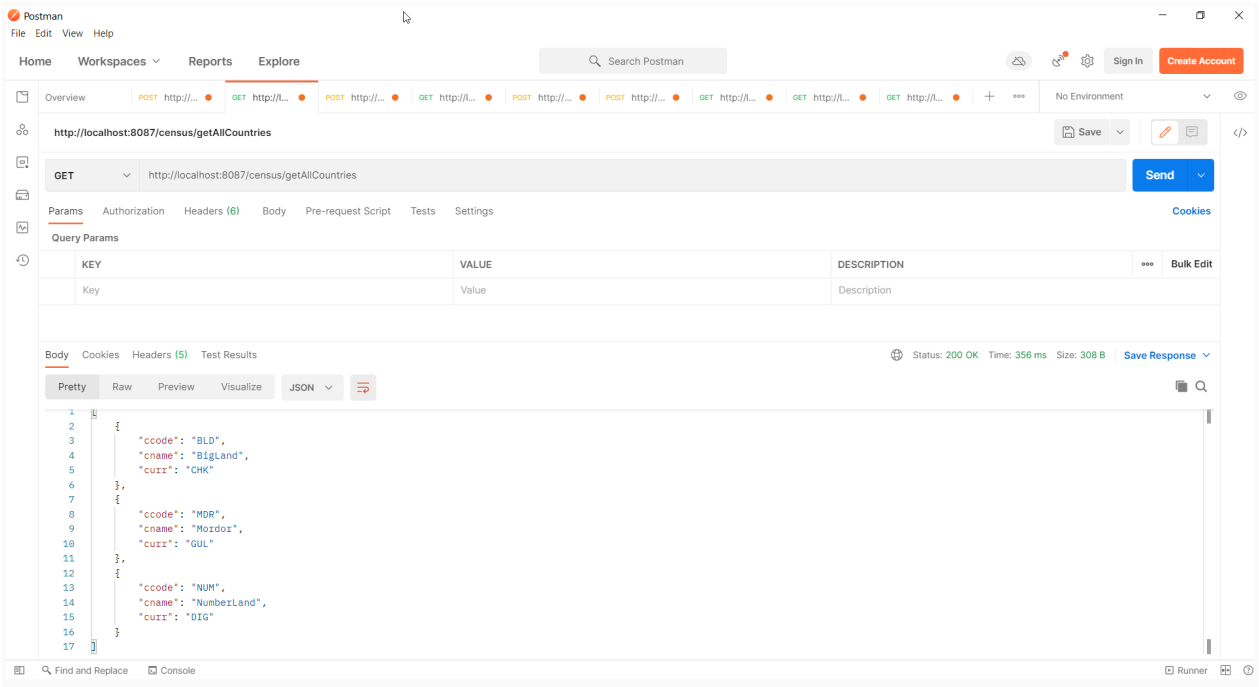
Likewise, the data is loaded

```
@Test
@Sql({"/import_data.sql"})
public void testLoadDataForTestCase() {
    assertEquals(3, stateRepository.findAll().size());
}
```

The application.properties file in Resources folder has the Mysql connectivity and server port configured.

```
spring.datasource.url=jdbc:mysql://localhost:3306/population
spring.datasource.username=root
spring.datasource.password=password
server.port = 8087
spring.jpa.hibernate.ddl-auto=none
```





`rest end points to be used (in postman) -`

`For adding a state -`

`http://localhost:8087/countries/`

In postman, in the Post, select body and json type and give the input -

```
{
    "ccode":"BLD",
    "stcode":"CTR",
    "stname":"CenterProvince",
    "pop":13987

}
```

For getting list of countries -

http://localhost:8087/countries/

adding state with post with given data as parmvalue-
http://localhost:8087/census/addState/MDR/STN/state/6964564

For retriving total population start a getrequest from postman -

http://localhost:8087/census/getAllCountryPop/NUM

where NUM is the country code

```
SELECT country.ccode, sum(state.pop)
                FROM population.country, population.state
                WHERE state.ccode = country.ccode and country.ccode =
```

For retrieving the validateState get request -

http://localhost:8087/census/validateState/BLD/RIG

where BLD is ccode and RIG is stcode

'BLD' ;

Query for validateState -

```
select count(*) from population.state where ccode =?1 and stcode =?2
```

Git Repository -

https://github.com/radhikakadiri/amex-demo-radhika.git
Census-demo.jar is the jar file that is also uploaded to git repository.