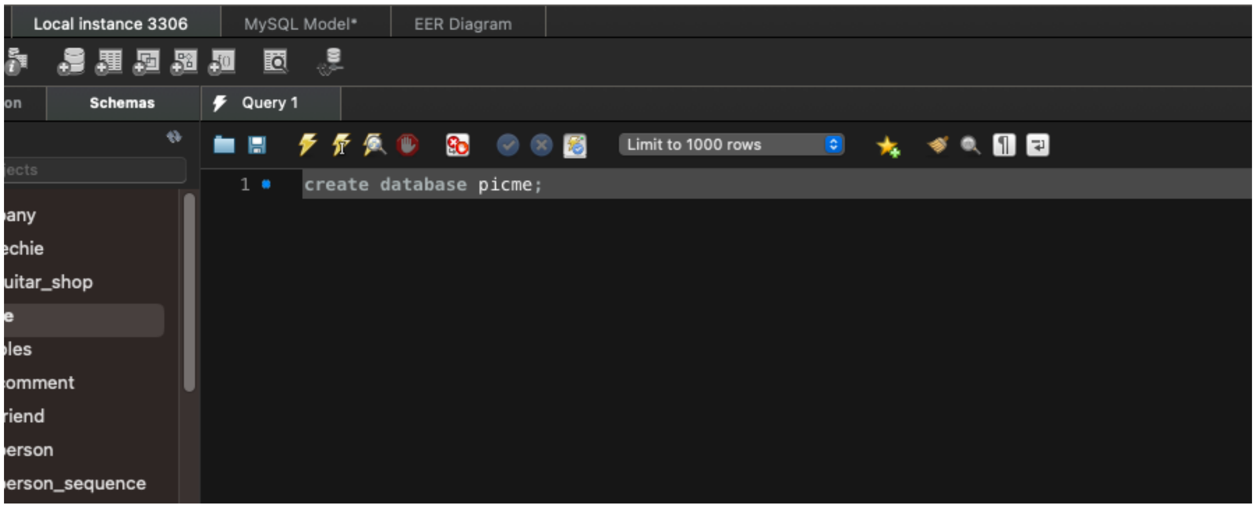
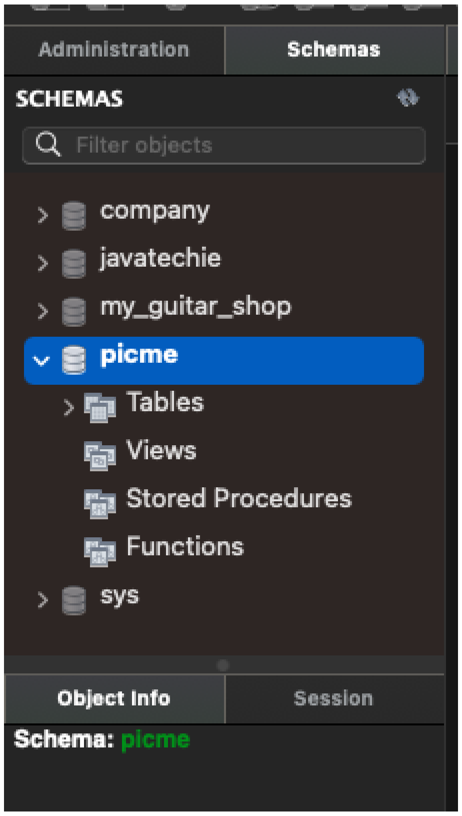
1. **Create the database in MySQL workbench**:
   - ➢ Open MySQL workbench and click on the local instance.
   - ➢ Write the following code into MySQL and click the lightning bolt.

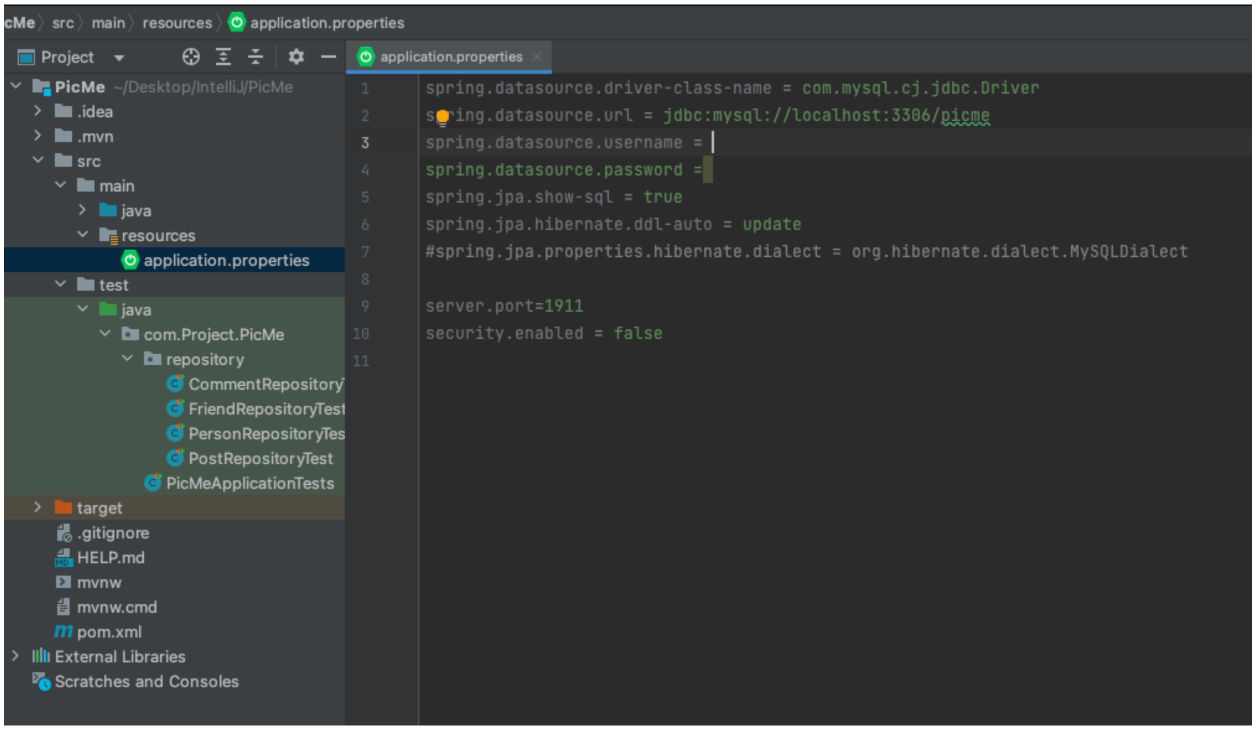


```
create database picme;
```

2. Then click on the refresh icon under the Schemas tab to see the new schema in the side panel.



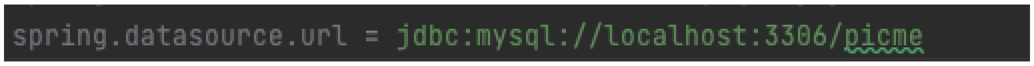
3. **Open IntelliJ and load up the PicMe project**:
   - ➢ Navigate to the application.properties file: (PicMe → src → main → resources)



```
spring.datasource.driver-class-name = com.mysql.cj.jdbc.Driver
spring.datasource.url = jdbc:mysql://localhost:3306/picme
spring.datasource.username =
spring.datasource.password =
spring.jpa.show-sql = true
spring.jpa.hibernate.ddl-auto = update
#spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQLDialect

server.port=1911
security.enabled = false
```

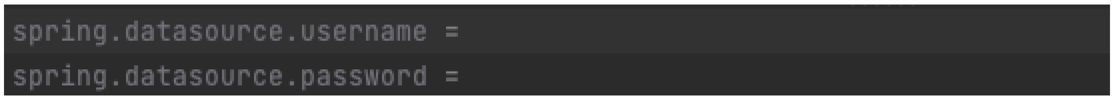
➢ Take note of the localhost port 3306. **If yours is different you will need to change that part but 3306 is the default**.

➢ After that is the “picme”, which is the name of the database we created earlier. Replace whatever word is there with “picme”

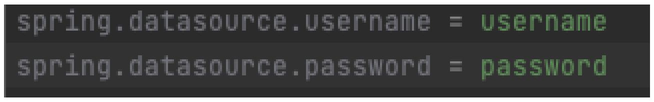


```
spring.datasource.url = jdbc:mysql://localhost:3306/picme
```

➢ After that is the username and password that you use for MySQL. You will have to enter your own username and password into the fields.



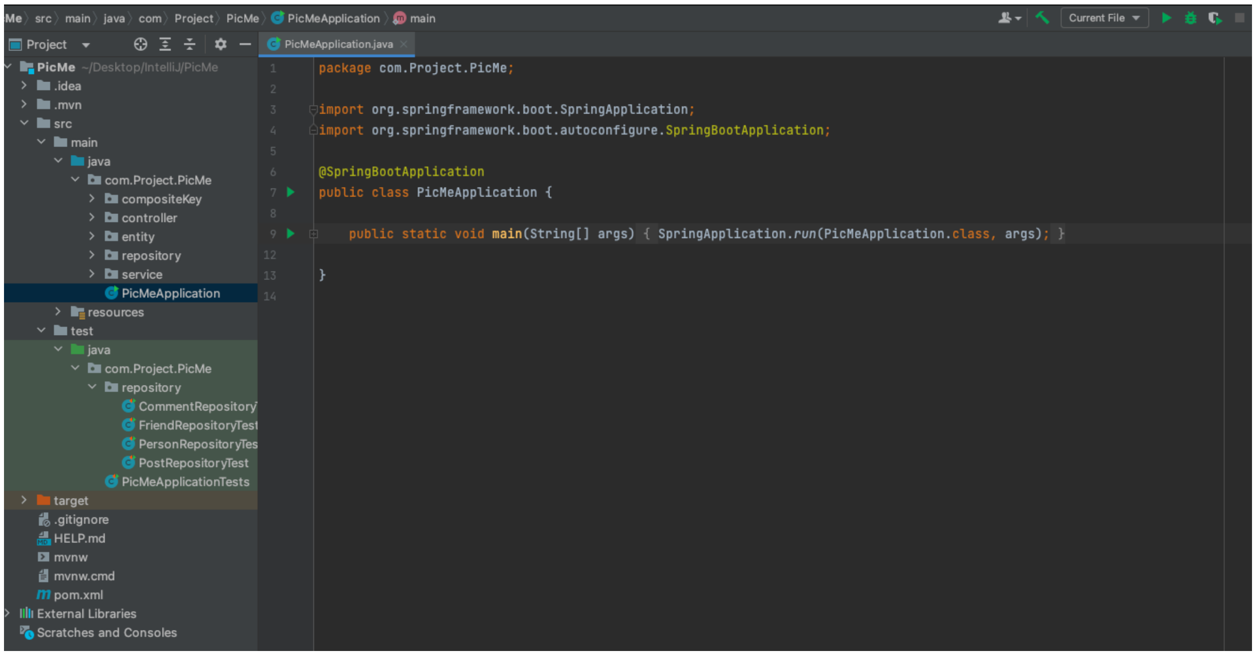
```
spring.datasource.username =
spring.datasource.password =
```



```
spring.datasource.username = username
spring.datasource.password = password
```

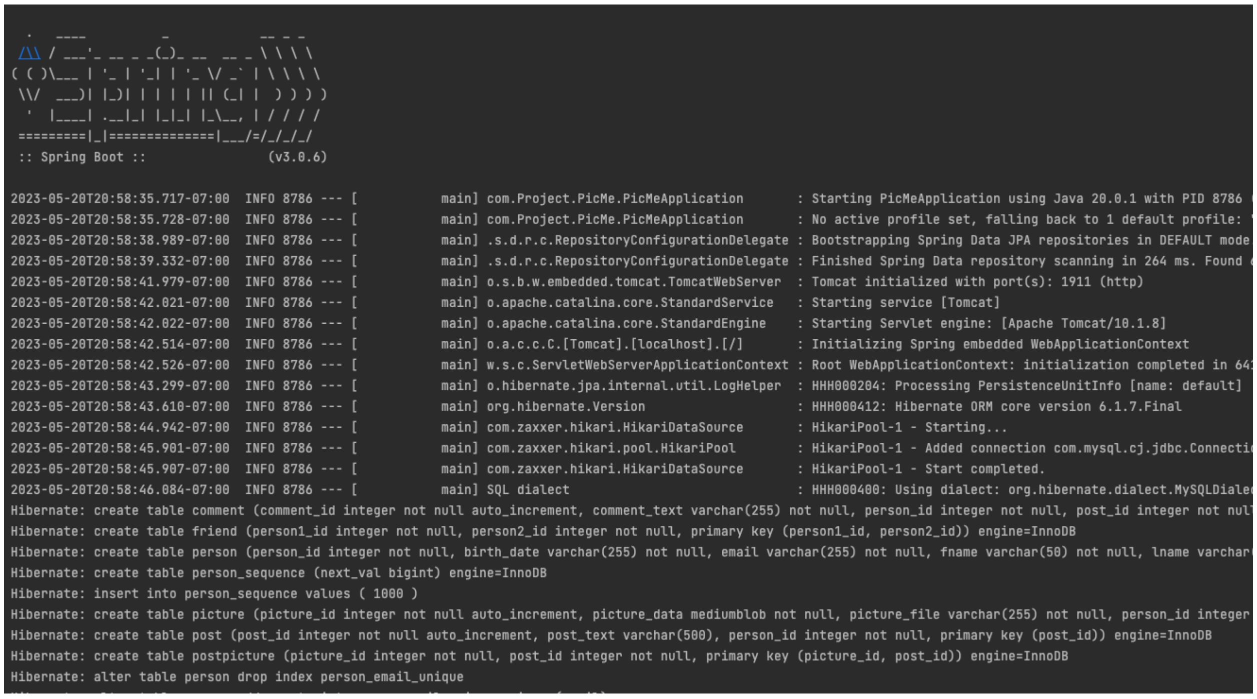
➢ After this step, you are done with setting up the link between the PicMe project and the database.

4. Now navigate to the PicMeApplication.java file and run it (PicMe → src → main → PicMeApplication)

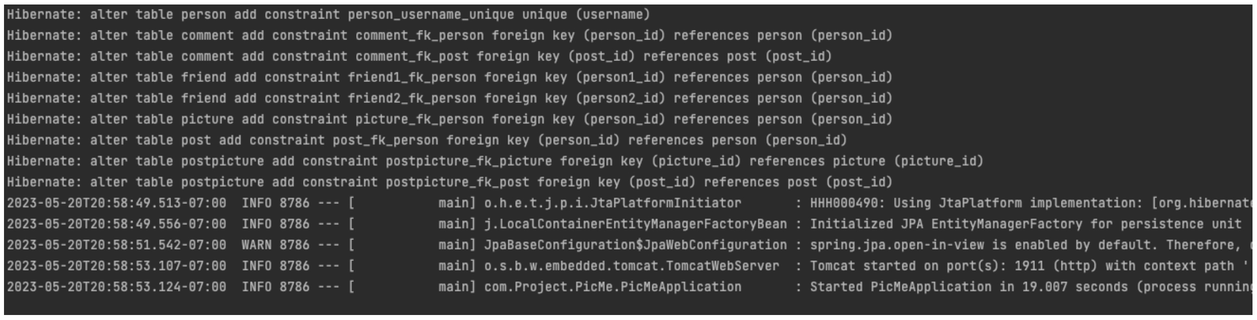


```java
package com.Project.PicMe;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class PicMeApplication {

    public static void main(String[] args) { SpringApplication.run(PicMeApplication.class, args); }

}
```

- ➢ It may take a while to start, but when it does finish the output to the console should look like this:



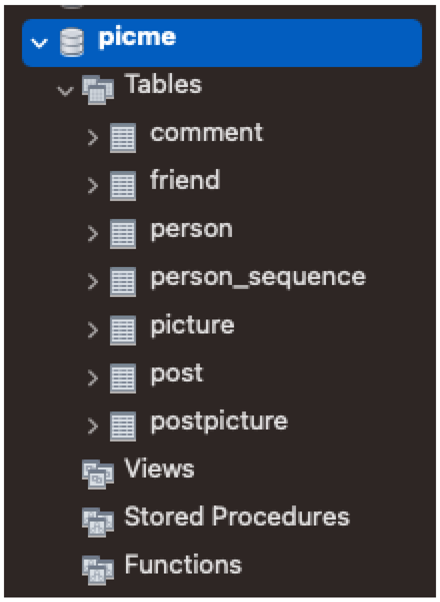
```
  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.0.6)

2023-05-20T20:58:35.717-07:00  INFO 8786 --- [           main] com.Project.PicMe.PicMeApplication       : Starting PicMeApplication using Java 20.0.1 with PID 8786
2023-05-20T20:58:35.728-07:00  INFO 8786 --- [           main] com.Project.PicMe.PicMeApplication       : No active profile set, falling back to 1 default profile:
2023-05-20T20:58:38.989-07:00  INFO 8786 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode
2023-05-20T20:58:39.332-07:00  INFO 8786 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 264 ms. Found
2023-05-20T20:58:41.979-07:00  INFO 8786 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s): 1911 (http)
2023-05-20T20:58:42.021-07:00  INFO 8786 --- [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2023-05-20T20:58:42.022-07:00  INFO 8786 --- [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat/10.1.8]
2023-05-20T20:58:42.514-07:00  INFO 8786 --- [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplicationContext
2023-05-20T20:58:42.526-07:00  INFO 8786 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 64
2023-05-20T20:58:43.299-07:00  INFO 8786 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
2023-05-20T20:58:43.610-07:00  INFO 8786 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 6.1.7.Final
2023-05-20T20:58:44.942-07:00  INFO 8786 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2023-05-20T20:58:45.901-07:00  INFO 8786 --- [           main] com.zaxxer.hikari.pool.HikariPool        : HikariPool-1 - Added connection com.mysql.cj.jdbc.Connecti
2023-05-20T20:58:45.907-07:00  INFO 8786 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2023-05-20T20:58:46.084-07:00  INFO 8786 --- [           main] SQL dialect                              : HHH000400: Using dialect: org.hibernate.dialect.MySQLDiale
Hibernate: create table comment (comment_id integer not null auto_increment, comment_text varchar(255) not null, person_id integer not null, post_id integer not nul
Hibernate: create table friend (person1_id integer not null, person2_id integer not null, primary key (person1_id, person2_id)) engine=InnoDB
Hibernate: create table person (person_id integer not null, birth_date varchar(255) not null, email varchar(255) not null, fname varchar(50) not null, lname varchar
Hibernate: create table person_sequence (next_val bigint) engine=InnoDB
Hibernate: insert into person_sequence values ( 1000 )
Hibernate: create table picture (picture_id integer not null auto_increment, picture_data mediumblob not null, picture_file varchar(255) not null, person_id integer
Hibernate: create table post (post_id integer not null auto_increment, post_text varchar(500), person_id integer not null, primary key (post_id)) engine=InnoDB
Hibernate: create table postpicture (picture_id integer not null, post_id integer not null, primary key (picture_id, post_id)) engine=InnoDB
Hibernate: alter table person drop index person_email_unique
```



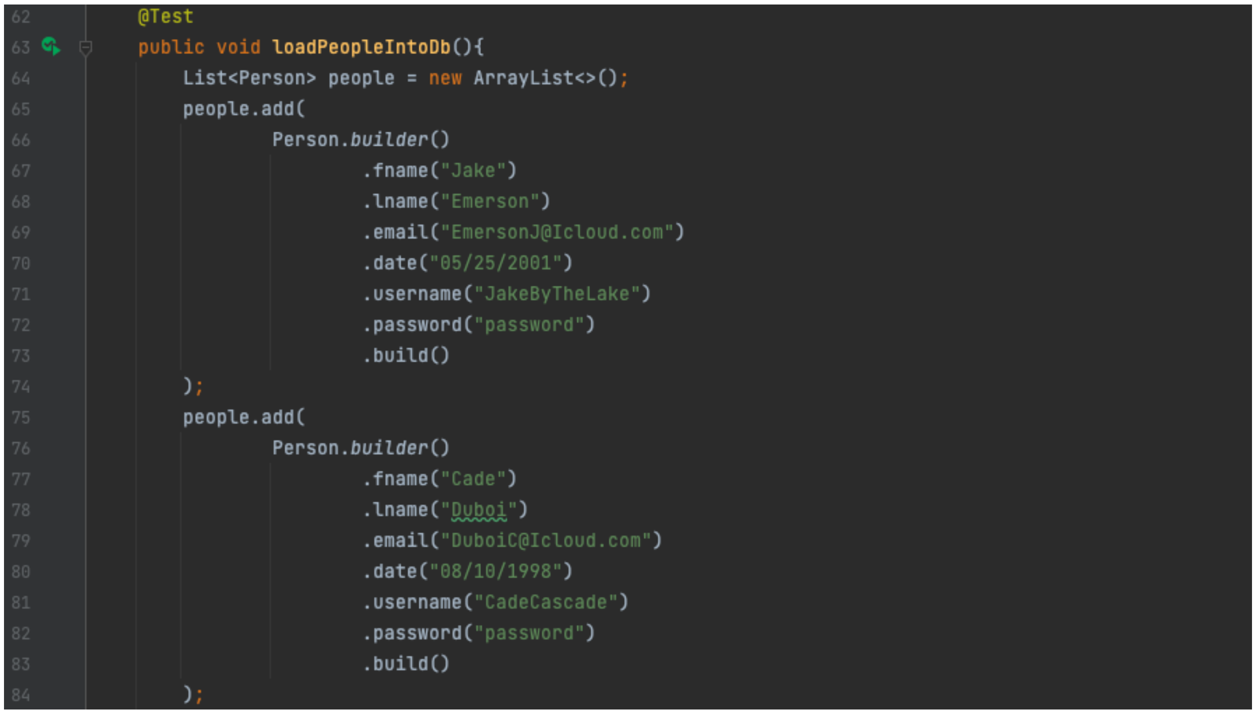
```
Hibernate: alter table person add constraint person_username_unique unique (username)
Hibernate: alter table comment add constraint comment_fk_person foreign key (person_id) references person (person_id)
Hibernate: alter table comment add constraint comment_fk_post foreign key (post_id) references post (post_id)
Hibernate: alter table friend add constraint friend1_fk_person foreign key (person1_id) references person (person_id)
Hibernate: alter table friend add constraint friend2_fk_person foreign key (person2_id) references person (person_id)
Hibernate: alter table picture add constraint picture_fk_person foreign key (person_id) references person (person_id)
Hibernate: alter table post add constraint post_fk_person foreign key (person_id) references person (person_id)
Hibernate: alter table postpicture add constraint postpicture_fk_picture foreign key (picture_id) references picture (picture_id)
Hibernate: alter table postpicture add constraint postpicture_fk_post foreign key (post_id) references post (post_id)
2023-05-20T20:58:49.513-07:00  INFO 8786 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000490: Using JtaPlatform implementation: [org.hibernat
2023-05-20T20:58:49.556-07:00  INFO 8786 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit
2023-05-20T20:58:51.542-07:00  WARN 8786 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore,
2023-05-20T20:58:53.107-07:00  INFO 8786 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 1911 (http) with context path '
2023-05-20T20:58:53.124-07:00  INFO 8786 --- [           main] com.Project.PicMe.PicMeApplication       : Started PicMeApplication in 19.007 seconds (process runnin
```

- ➢ You should see all of the SQL statements that were sent to the database.
- ➢ If you see a bunch of errors, something has gone wrong and you will have to troubleshoot as there could be multiple reasons for it.

5. Return to the MySQL workbench pic me schema and use the refresh icon once again to see the changes in the database:
   - ➢ Click on the drop-down menu for tables and you should now see all of the tables loaded in.

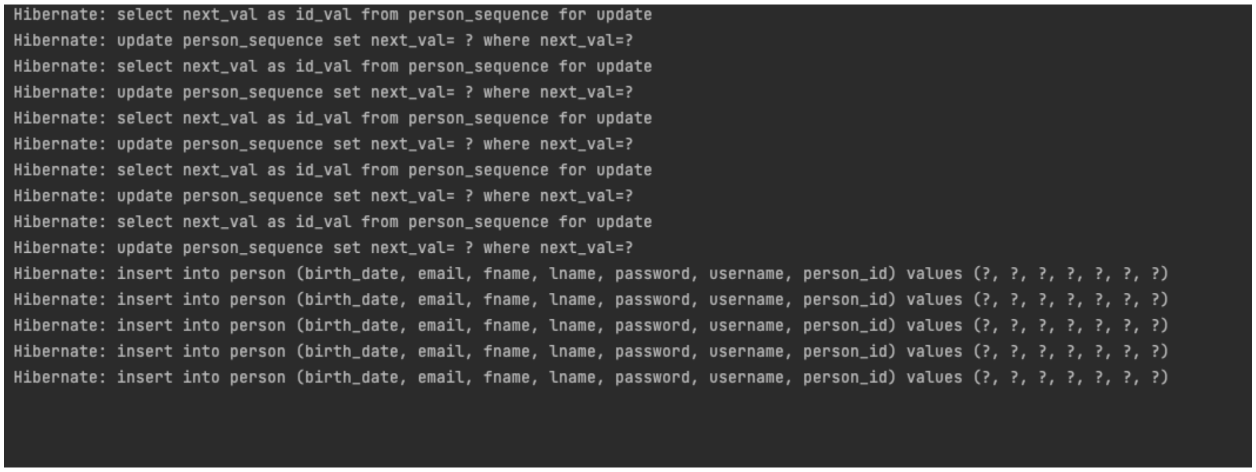


6. Load test users into the database:
   - ➢ Go back to the PicMe project and navigate to the PersonRepositoryTest class. (PicMe → src → test → Java → com.Project.PicMe → repository → PersonRepository)
   - ➢ Scroll down until you see the loadPeopleIntoDb test function. Click on the green play button to run it.



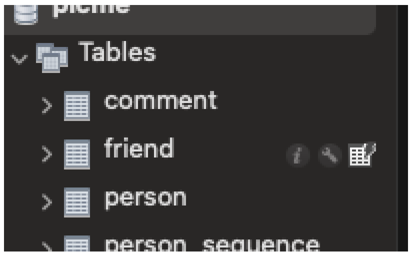
```java
    @Test
    public void loadPeopleIntoDb(){
        List<Person> people = new ArrayList<>();
        people.add(
                Person.builder()
                        .fname("Jake")
                        .lname("Emerson")
                        .email("EmersonJ@Icloud.com")
                        .date("05/25/2001")
                        .username("JakeByTheLake")
                        .password("password")
                        .build()
        );
        people.add(
                Person.builder()
                        .fname("Cade")
                        .lname("Duboi")
                        .email("DuboiC@Icloud.com")
                        .date("08/10/1998")
                        .username("CadeCascade")
                        .password("password")
                        .build()
        );
```

➢ You will see similar console output as before then these SQL statements should flash out to the console.

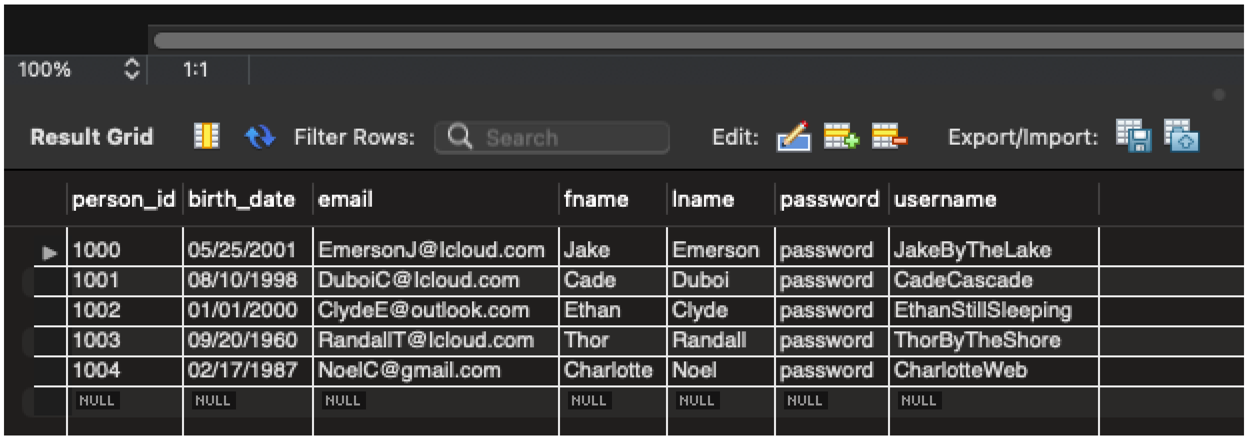


```
Hibernate: select next_val as id_val from person_sequence for update
Hibernate: update person_sequence set next_val= ? where next_val=?
Hibernate: select next_val as id_val from person_sequence for update
Hibernate: update person_sequence set next_val= ? where next_val=?
Hibernate: select next_val as id_val from person_sequence for update
Hibernate: update person_sequence set next_val= ? where next_val=?
Hibernate: select next_val as id_val from person_sequence for update
Hibernate: update person_sequence set next_val= ? where next_val=?
Hibernate: select next_val as id_val from person_sequence for update
Hibernate: update person_sequence set next_val= ? where next_val=?
Hibernate: insert into person (birth_date, email, fname, lname, password, username, person_id) values (?, ?, ?, ?, ?, ?, ?)
Hibernate: insert into person (birth_date, email, fname, lname, password, username, person_id) values (?, ?, ?, ?, ?, ?, ?)
Hibernate: insert into person (birth_date, email, fname, lname, password, username, person_id) values (?, ?, ?, ?, ?, ?, ?)
Hibernate: insert into person (birth_date, email, fname, lname, password, username, person_id) values (?, ?, ?, ?, ?, ?, ?)
Hibernate: insert into person (birth_date, email, fname, lname, password, username, person_id) values (?, ?, ?, ?, ?, ?, ?)
```

7. View changes in the database:
    - ➢ Open back up MySQL workbench and find the person table. Hover over it and click on the gridded box icon.



    - ➢ The data in the table should show on the bottom of the screen.



person_id	birth_date	email	fname	lname	password	username
1000	05/25/2001	EmersonJ@Icloud.com	Jake	Emerson	password	JakeByTheLake
1001	08/10/1998	DuboiC@Icloud.com	Cade	Duboi	password	CadeCascade
1002	01/01/2000	ClydeE@outlook.com	Ethan	Clyde	password	EthanStillSleeping
1003	09/20/1960	RandallT@Icloud.com	Thor	Randall	password	ThorByTheShore
1004	02/17/1987	NoelC@gmail.com	Charlotte	Noel	password	CharlotteWeb
NULL	NULL	NULL	NULL	NULL	NULL	NULL

As of the time that I'm making this the PersonRepositoryTest is the only test class that can load in multiple tuples. However, the other RepositoryTest classes have functions to load in single entities at a time.