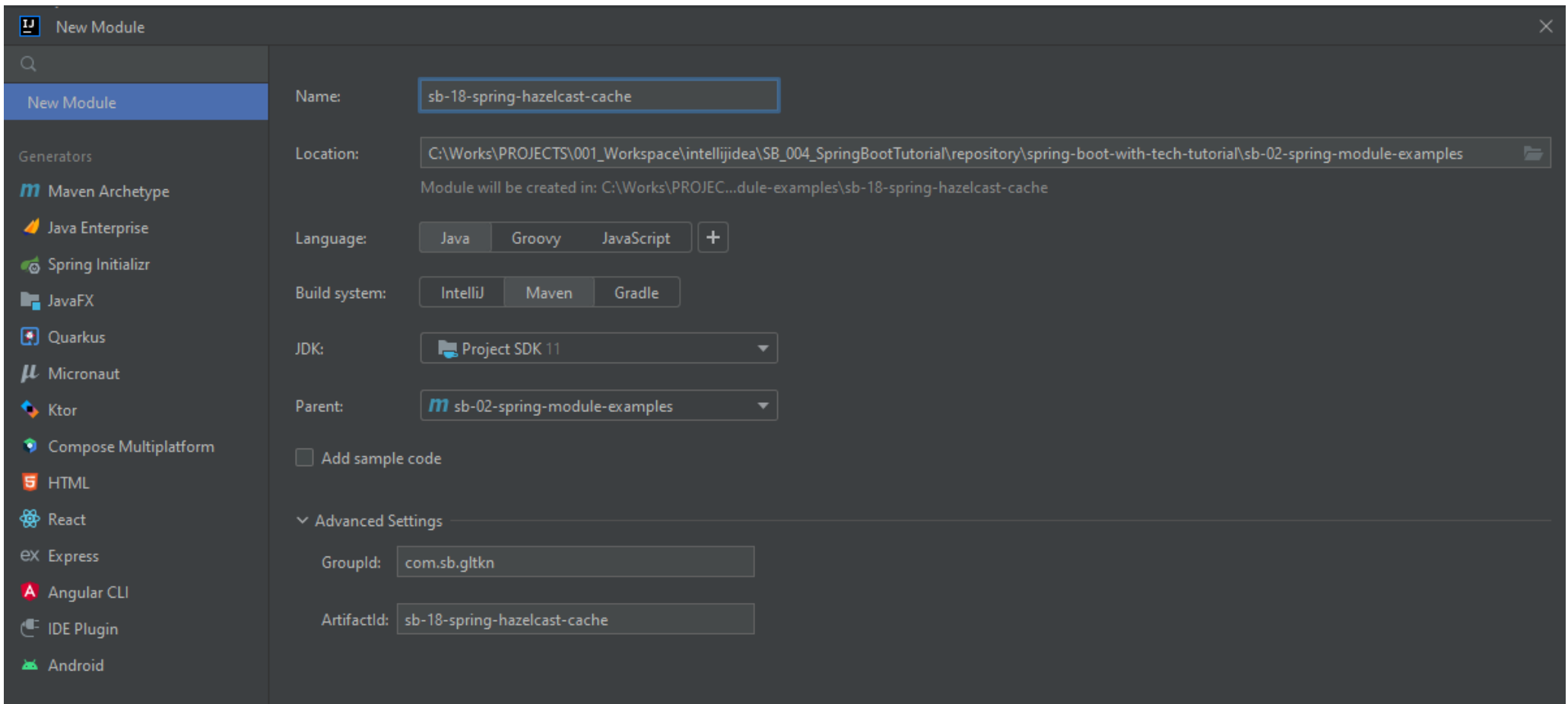
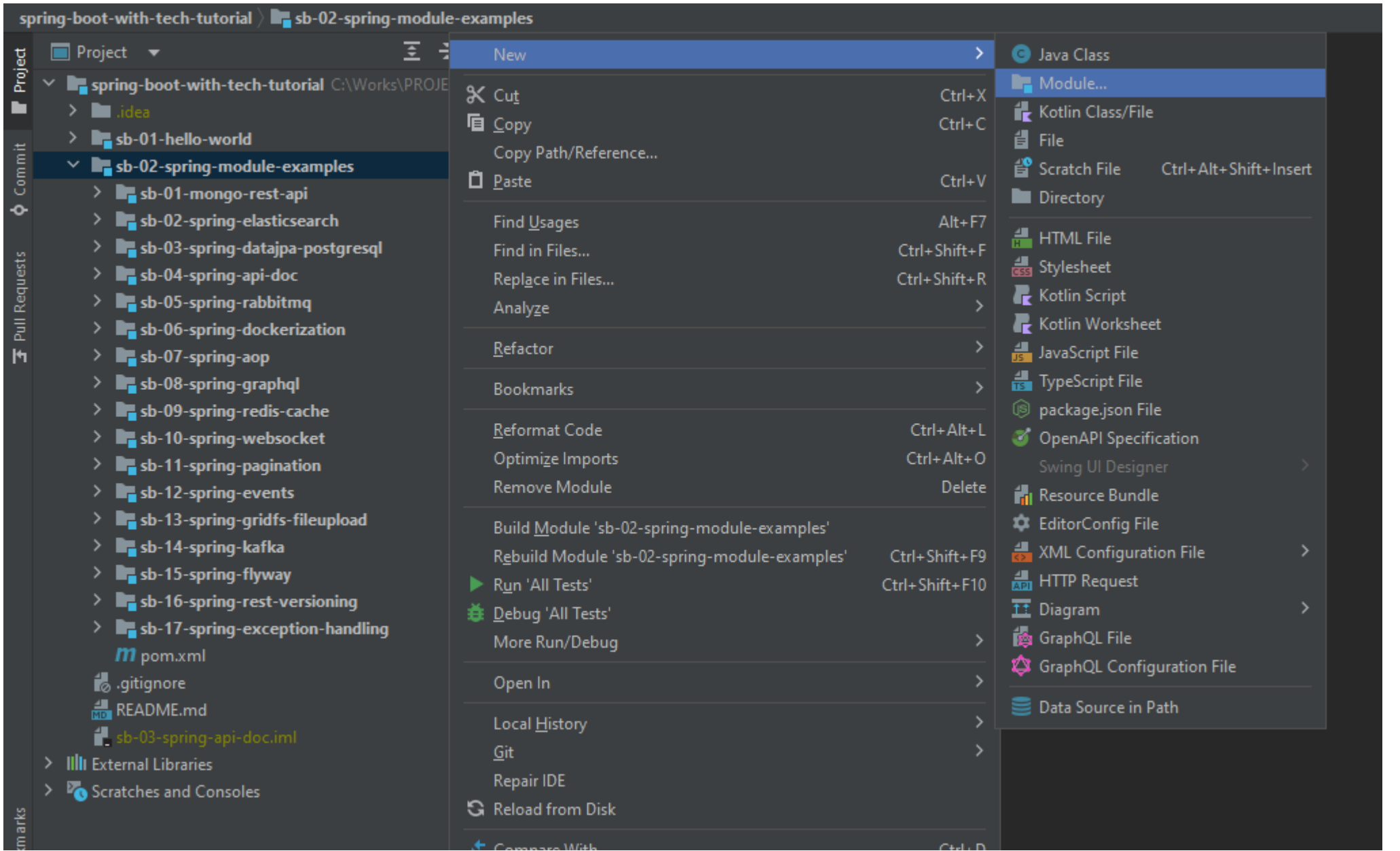


Hazelcast ile hibernate cache'i kullanacağımız bir spring boot uygulaması yapalım.



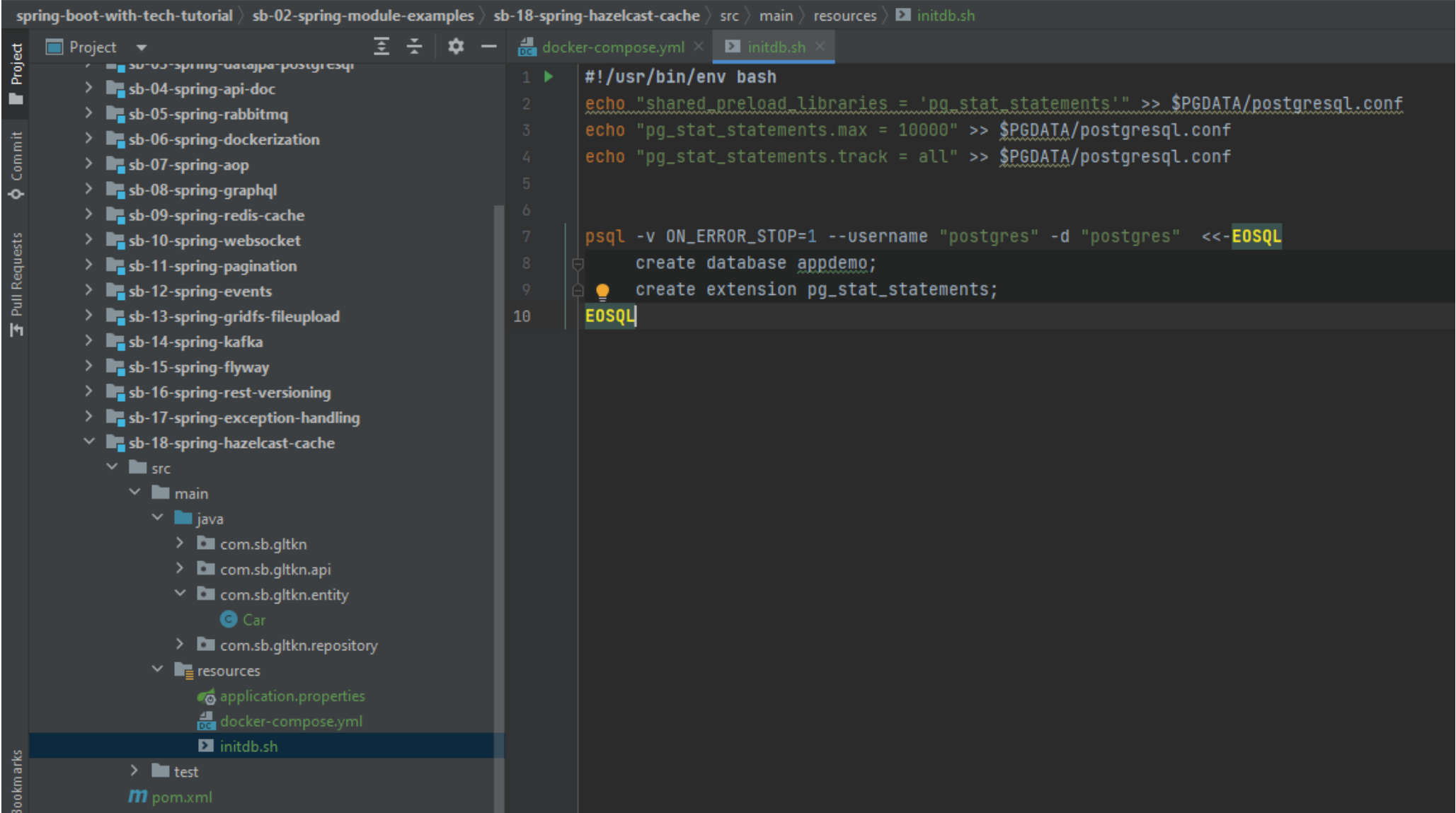
```
spring-boot-with-tech-tutorial > sb-02-spring-module-examples > sb-18-spring-hazelcast-cache > pom.xml
pom.xml (sb-18-spring-hazelcast-cache)
19 <dependencies>
20 <dependency>
21 <groupId>org.springframework.boot</groupId>
22 <artifactId>spring-boot-starter-data-jpa</artifactId>
23 </dependency>
24 <dependency>
25 <groupId>org.springframework.boot</groupId>
26 <artifactId>spring-boot-starter-web</artifactId>
27 </dependency>
28 <dependency>
29 <groupId>org.postgresql</groupId>
30 <artifactId>postgresql</artifactId>
31 <scope>runtime</scope>
32 </dependency>
33 <dependency>
34 <groupId>org.projectlombok</groupId>
35 <artifactId>lombok</artifactId>
36 <optional>true</optional>
37 </dependency>
38 <dependency>
39 <groupId>com.hazelcast</groupId>
40 <artifactId>hazelcast-spring</artifactId>
41 </dependency>
42 <dependency>
43 <groupId>com.hazelcast</groupId>
44 <artifactId>hazelcast-hibernate53</artifactId>
45 <version>2.0.0</version>
46 </dependency>
47 <dependency>
48 <groupId>org.hibernate</groupId>
49 <artifactId>hibernate-jcache</artifactId>
50 </dependency>
51 </dependencies>
```

docker-compose ile veritabanı sunucusu ayağı kalkmadan önce **initdb.sh** çalıştırılacaktır.

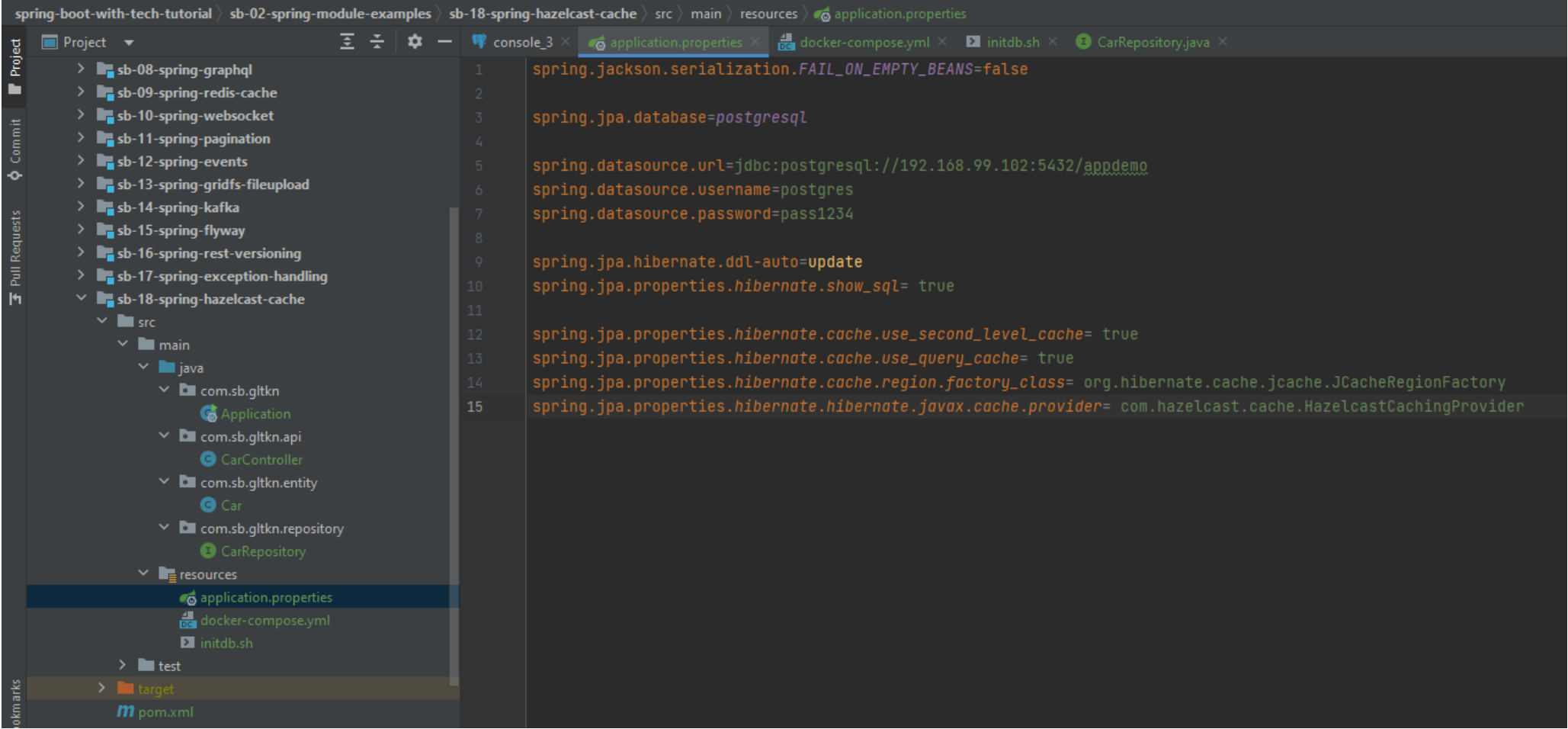
```
spring-boot-with-tech-tutorial > sb-02-spring-module-examples > sb-18-spring-hazelcast-cache > src > main > resources > docker-compose.yml
docker-compose.yml
1 # Use postgres/example user/password credentials
2 # https://www.postgresql.org/docs/9.4/pgstatstatements.html
3
4 version: '3.1'
5 services:
6   db:
7     image: postgres
8     restart: always
9     volumes:
10      - ./initdb.sh:/docker-entrypoint-initdb.d/initdb.sh
11     ports:
12      - 5432:5432
13     environment:
14      POSTGRES_PASSWORD: postgres
```

pg_stat_statements adlı bir extension'ını kullanarak, veritabanının işlediği sorgular hakkında birçok şeyi belirlemek için hesaplamaları kullanabiliriz. **pg_stat_statements** extension'ını kullanmadan önce, bunu konfigürasyon dosyanızda etkinleştirmemiz ve örneğimizi yeniden başlatmanız gerekir.

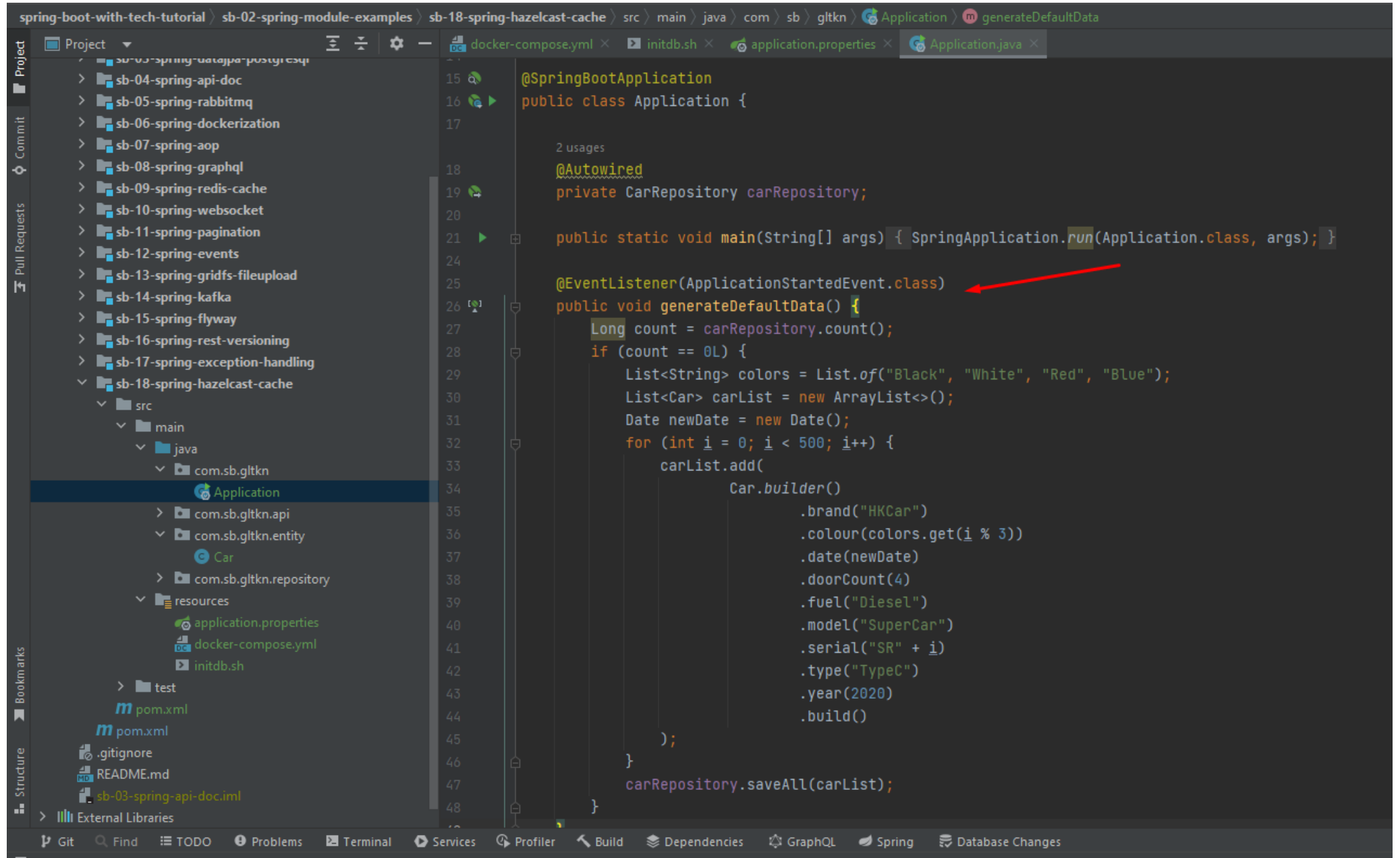
Postgresql database'ine atılan tüm sorguları log'layan **pg_stat_statements**'i etkinleştiriyoruz.
Maximum 10.000 statement tutulmasını bildirdik.
appdemo veritabanı oluşturulacak.



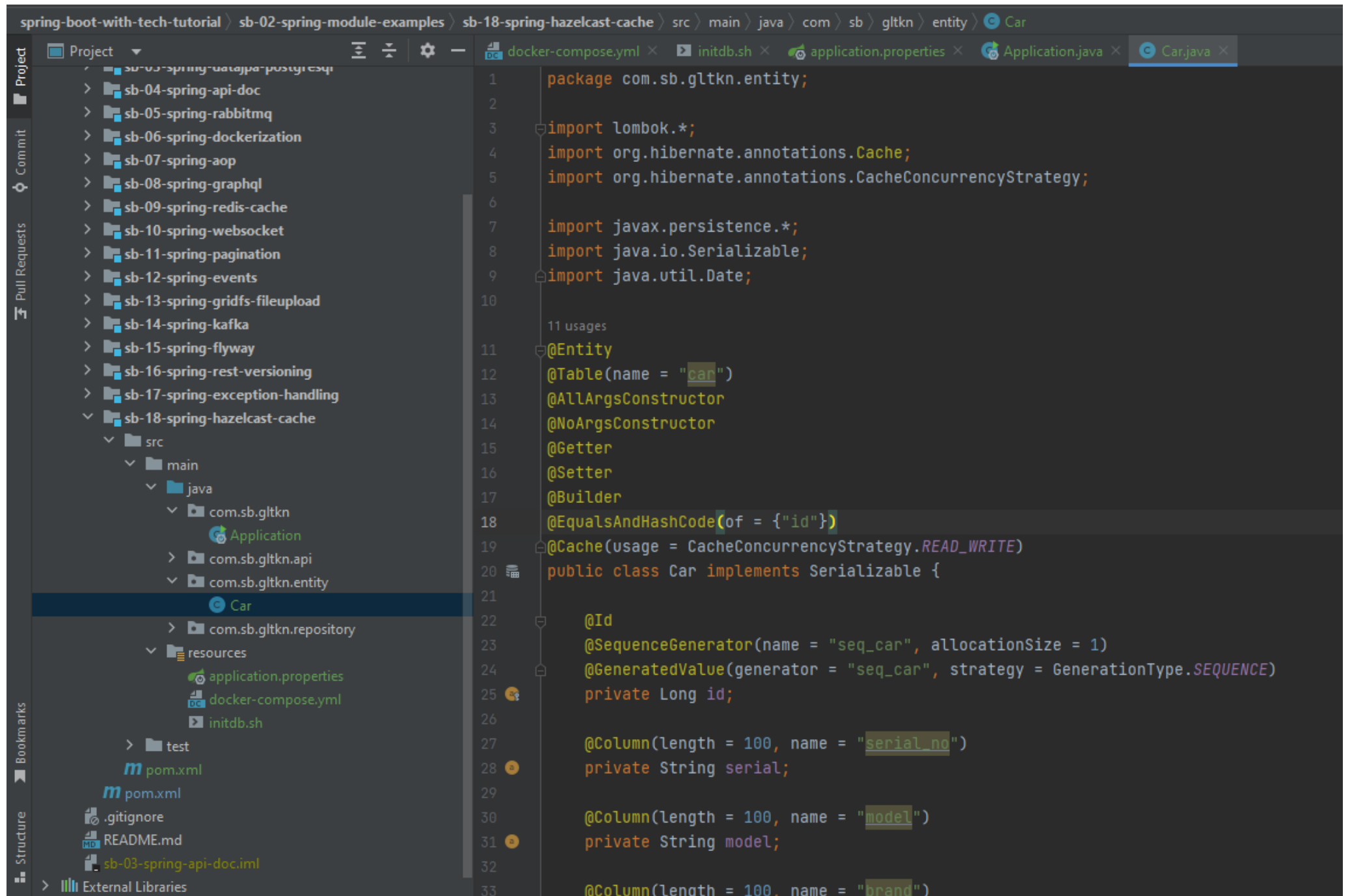
Aşağıdaki konfigürasyon ile aşağıda hazelcastCaching özelliğini aktif hale getiriyoruz.
Hibernate second level cache'i enable ediyoruz.
Query cache'i enable ediyoruz.



generatedDefaultData metodu uygulama ayağı kalktığında devreye girecek ve veritabanında data bulunamazsa 500 veriyi veritabanına kaydedecektir.

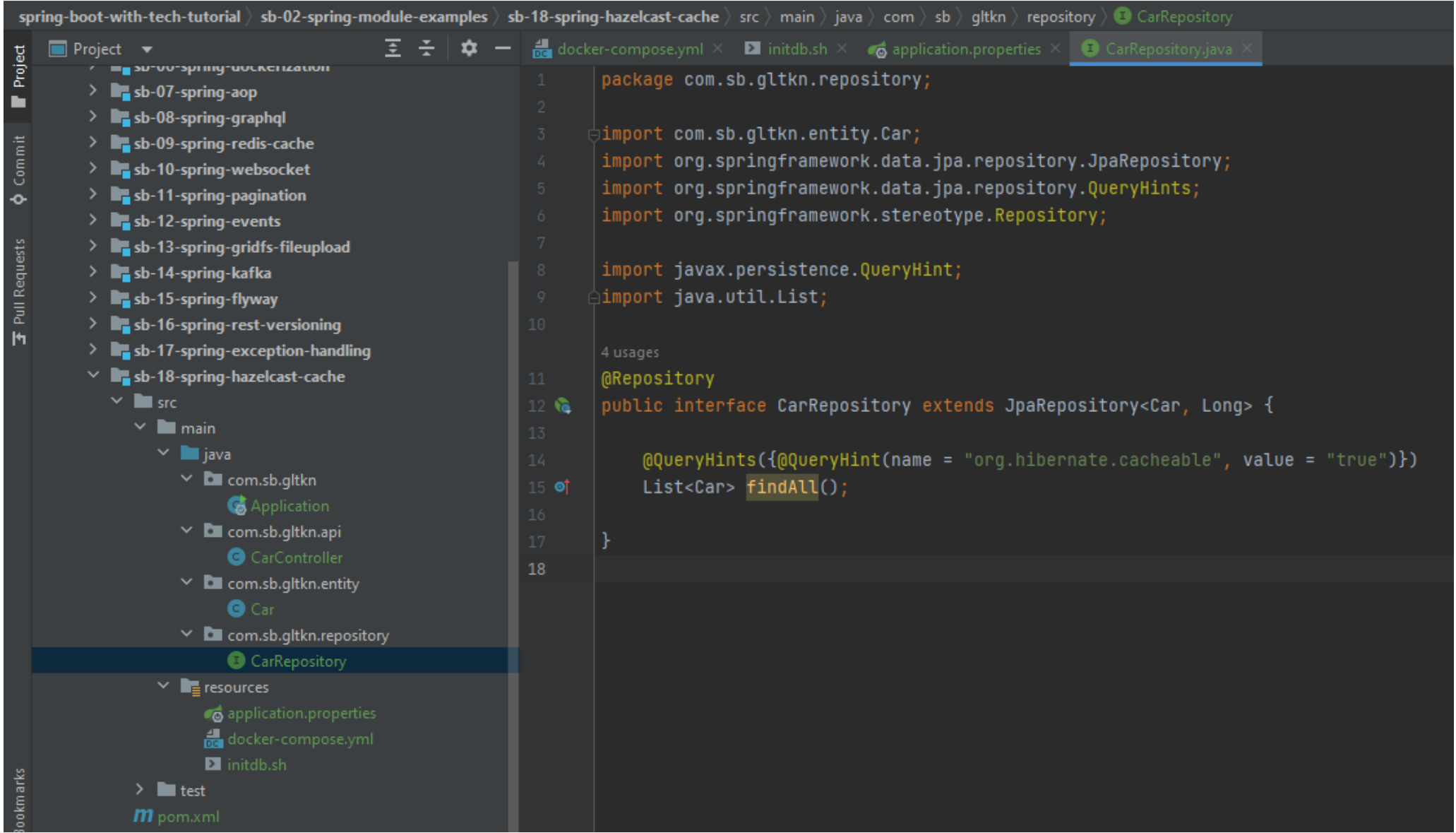


```
15 @SpringBootApplication
16 public class Application {
17
18     2 usages
19     @Autowired
20     private CarRepository carRepository;
21
22     public static void main(String[] args) { SpringApplication.run(Application.class, args); }
23
24     @EventListener(ApplicationStartedEvent.class)
25     public void generateDefaultData() {
26         Long count = carRepository.count();
27         if (count == 0L) {
28             List<String> colors = List.of("Black", "White", "Red", "Blue");
29             List<Car> carList = new ArrayList<>();
30             Date newDate = new Date();
31             for (int i = 0; i < 500; i++) {
32                 carList.add(
33                     Car.builder()
34                         .brand("HKCar")
35                         .colour(colors.get(i % 3))
36                         .date(newDate)
37                         .doorCount(4)
38                         .fuel("Diesel")
39                         .model("SuperCar")
40                         .serial("SR" + i)
41                         .type("TypeC")
42                         .year(2020)
43                         .build()
44                 );
45             }
46             carRepository.saveAll(carList);
47         }
48     }
49 }
```



```
1 package com.sb.gltkn.entity;
2
3 import lombok.*;
4 import org.hibernate.annotations.Cache;
5 import org.hibernate.annotations.CacheConcurrencyStrategy;
6
7 import javax.persistence.*;
8 import java.io.Serializable;
9 import java.util.Date;
10
11 11 usages
12 @Entity
13 @Table(name = "car")
14 @AllArgsConstructor
15 @NoArgsConstructor
16 @Getter
17 @Setter
18 @Builder
19 @EqualsAndHashCode(of = {"id"})
20 @Cache(usage = CacheConcurrencyStrategy.READ_WRITE)
21 public class Car implements Serializable {
22
23     @Id
24     @SequenceGenerator(name = "seq_car", allocationSize = 1)
25     @GeneratedValue(generator = "seq_car", strategy = GenerationType.SEQUENCE)
26     private Long id;
27
28     @Column(length = 100, name = "serial_no")
29     private String serial;
30
31     @Column(length = 100, name = "model")
32     private String model;
33
34     @Column(length = 100, name = "brand")
35     private String brand;
36 }
```


Aşağıdaki gibi findAll sorgusu cache'lenecektir.



```
EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker-machine ls
NAME      ACTIVE  DRIVER        STATE     URL                  SWARM   DOCKER        ERRORS
default   -       virtualbox    Running   tcp://192.168.99.102:2376   v19.03.12
```

```
EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker-machine env default
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://192.168.99.102:2376"
export DOCKER_CERT_PATH="C:\Users\EmreGltkn\.docker\machine\machines\default"
export DOCKER_MACHINE_NAME="default"
export COMPOSE_CONVERT_WINDOWS_PATHS="true"
# Run this command to configure your shell:
# eval "$(C:\Users\EmreGltkn\bin\docker-machine.exe env default)"

EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ eval "$(docker-machine env default)"
```

docker-compose -f /c/Works/PROJECTS/001_Workspace/intellijidea/SB_004_SpringBootTutorial/repository/spring-boot-with-tech-tutorial/sb-02-spring-module-examples/sb-18-spring-hazelcast-cache/src/main/resources/docker-compose.yml up -d

```
EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker-compose -f /c/Works/PROJECTS/001_Workspace/intellijidea/SB_004_SpringBootTutorial/repository/spring-boot-with-tech-tutorial/sb-02-spring-module-examples/sb-18-spring-hazelcast-cache/src/main/resources/docker-compose.yml up -d
Found orphan containers (gltkn-kafka) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans flag to clean it up.
Recreating resources_db_1 ...
Recreating resources_db_1 ... done

EmreGltkn@DESKTOP-DN9PH1A MINGW64 ~
$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
0d967bede788   postgres  "docker-entrypoint.s..." 11 seconds ago Up 4 seconds  0.0.0.0:5432->5432/tcp             resources_db_1
```

Project

sb-03-spring-api-doc.iml

External Libraries

Scratches and Consoles

Database Consoles

postgres@192.168.99.102

console

console_1

console_2

console_3

Extensions

Database Tools and SQL

Jakarta EE: Persistence (JPA)

data

orm.generators

schema

schema.layouts

predefinedExternalAnnotations.json

console_3

application.properties

CarRepository.java

create database appdemo;

create extension pg_stat_statements;

select * from pg_stat_statements;

Database

postgres@192.168.99.102

appdemo

public

Database Objects

access methods

casts

extensions

pg_stat_statements

plpgsql

languages

postgres

testdb

Server Objects

Services

Spring Boot

Not Started

postgres@192.168.99.102

console_3 18 ms

Output

appdemo.public.pg_stat_statements

273 rows

userid

dbid

toplevel

queryid

query

plans

total_plan_t

262	10	13757	true	5912361476795981438	SET application_name = 'IntelliJ IDEA 2022.1.1'	0	
263	10	16394	true	6781833850498940997	select oid from pg_catalog.pg_opfamily where opfname...	0	
264	10	16394	true	2559325606635865253	select oid as id, umserver as server_id, ...	0	
265	10	13757	true	-4782613137476937854	select l.oid as id, l.xmin state_number, lanname as ...	0	
266	10	16394	true	-312787843980422199	SELECT e.typdelim FROM pg_catalog.pg_type t, pg_cata...	0	
267	10	13757	true	8863309718139044683	select T.relkind as view_kind, T.oid a...	0	
268	10	24596	true	2639914821738652065	select round(extract(\$1 from pg_postmaster_start_tim...	0	
269	10	16394	true	383640018708525686	select tab.oid table_id, tab.re...	0	
270	10	16394	true	1773839100117194729	select 0.oid as id, 0.amopstrategy as strateg...	0	