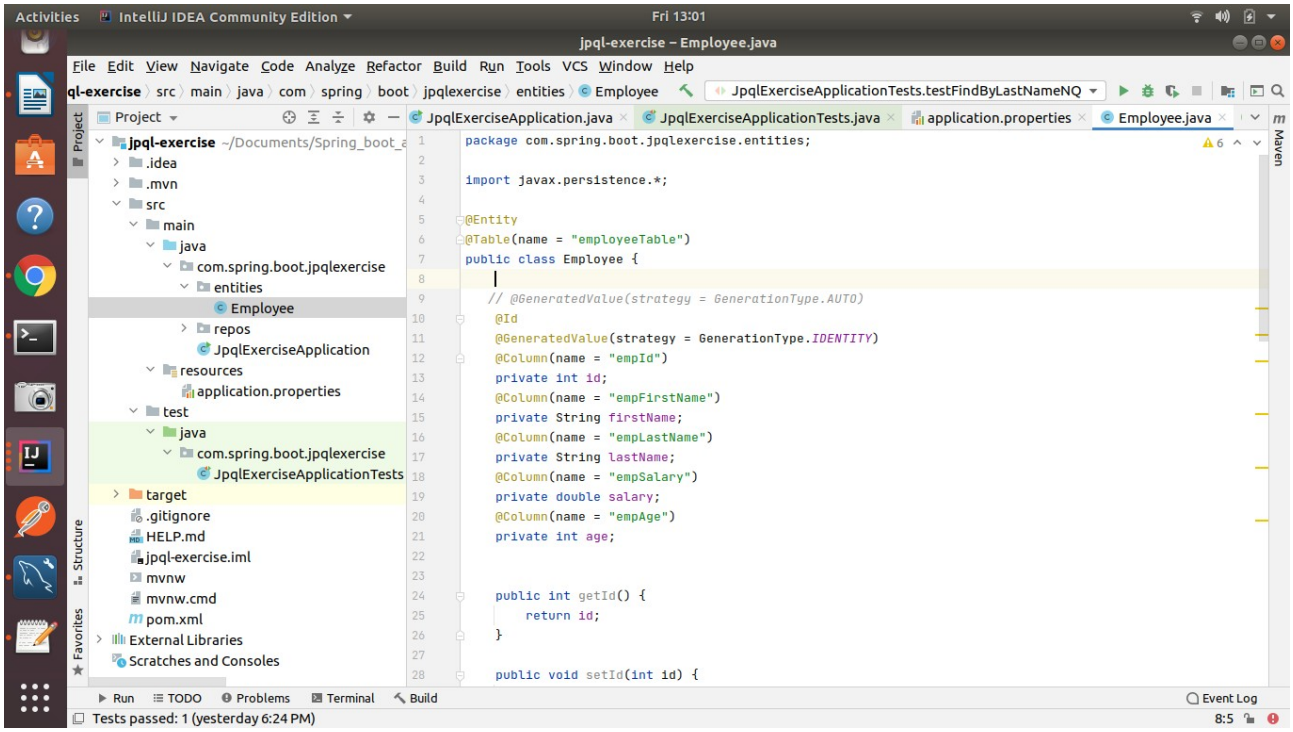
Spring Data JPA with Hibernate Part 2 Exercise

JPQL:

1. Display the first name, last name of all employees having salary greater than average salary ordered in ascending by their age and in descending by their salary.



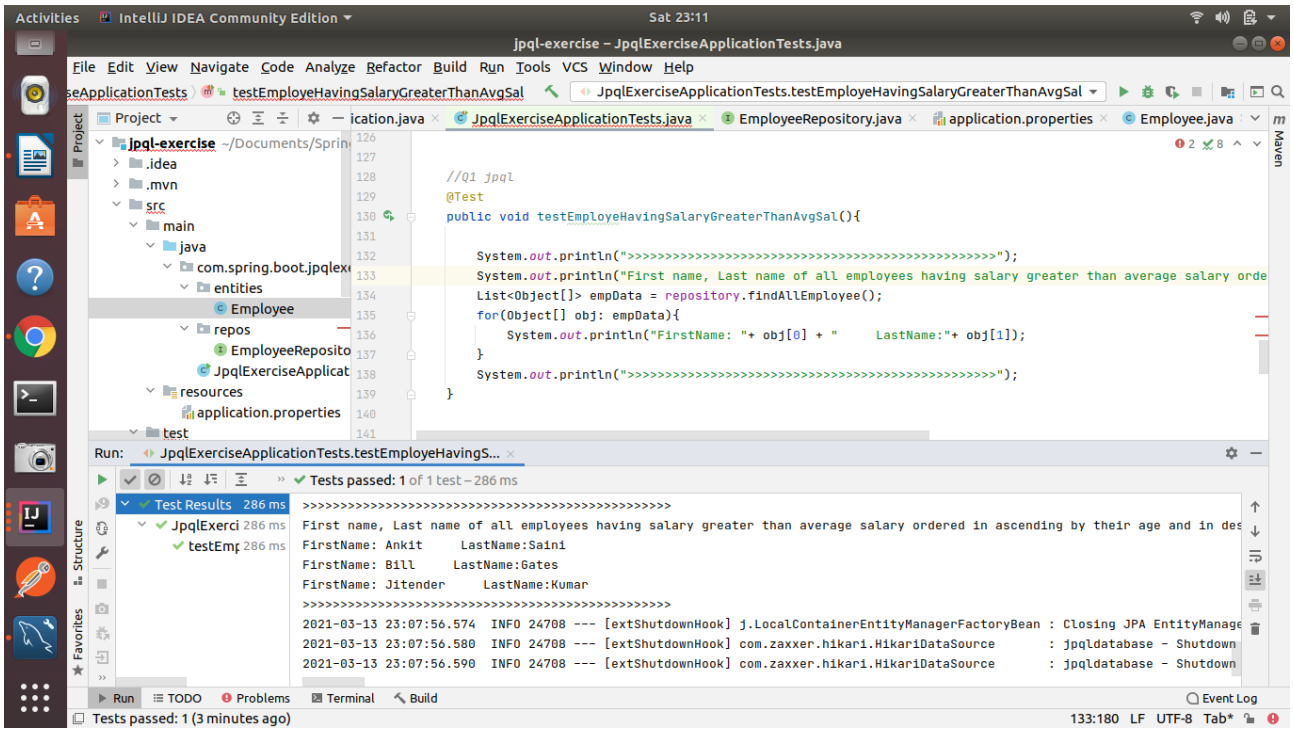

```java
package com.spring.boot.jpqlexercise.repos;

import com.spring.boot.jpqlexercise.entities.Employee;
import org.springframework.data.jpa.repository.Modifying;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.PagingAndSortingRepository;
import org.springframework.data.repository.query.Param;

import javax.transaction.Transactional;
import java.util.List;

public interface EmployeeRepository extends CrudRepository<Employee, Integer> {

    //Q1 jpql
    @Query("select firstName, lastName from Employee where " +
            "salary > (select AVG(salary) from Employee)" +
            " ORDER BY age ASC, salary DESC")
    List<Object[]> findAllEmployee();
```



2. Update salary of all employees by a salary passed as a parameter whose existing salary is less than the average salary.

```
//Ques2 jpql
@Modifying
@Transactional
@Query("update Employee e set e.salary=:salary where e.salary< (select AVG(ee.salary) from Employee ee)")
void findAllEmployeesUpdateSalary(@Param("salary") double salary);
```

```
//Q2 jpql
@Test
//@Transactional
//@Rollback(value = false)
public void testFindAllEmployeesUpdateSalary(){
    double salary = 50000;
    repository.findAllEmployeesUpdateSalary(salary);
}
```

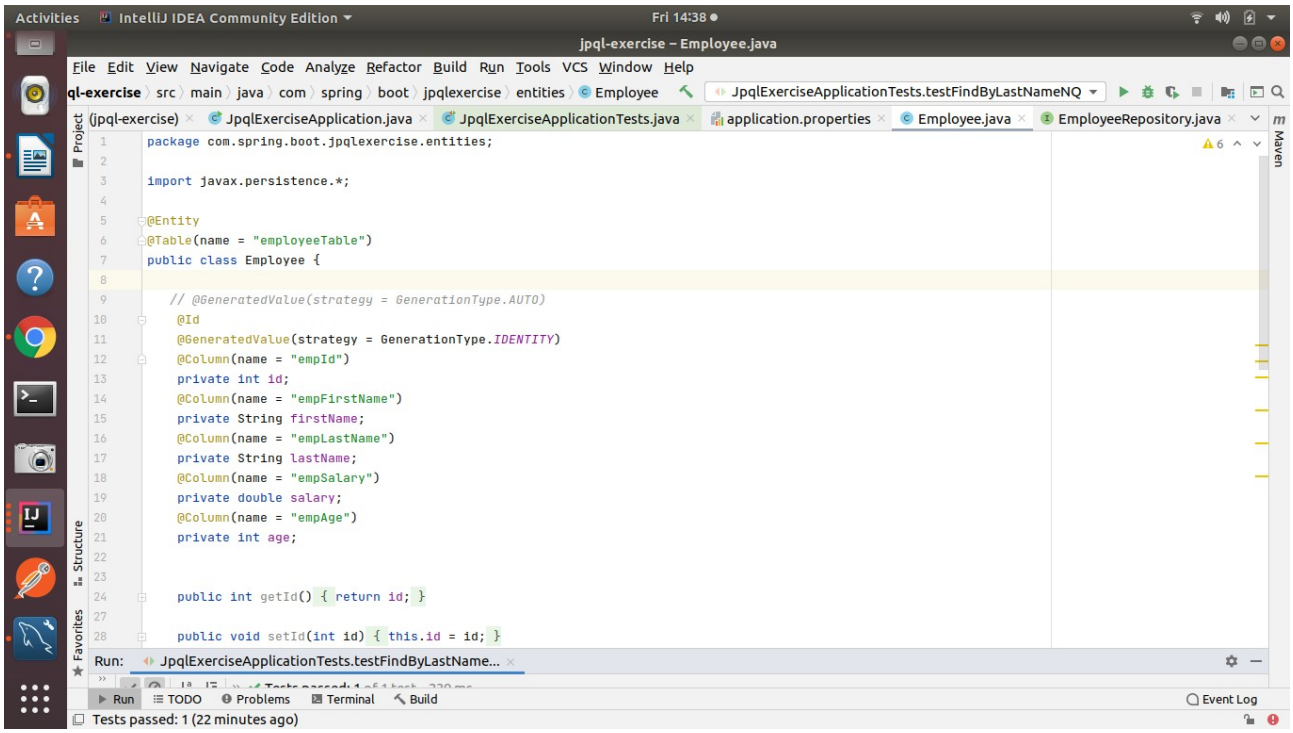
3. Delete all employees with minimum salary.

```java
//Q3 jpql
@Modifying
@Query("DELETE FROM Employee WHERE salary < (select AVG(salary) FROM Employee)")
void deleteEmployees();
```

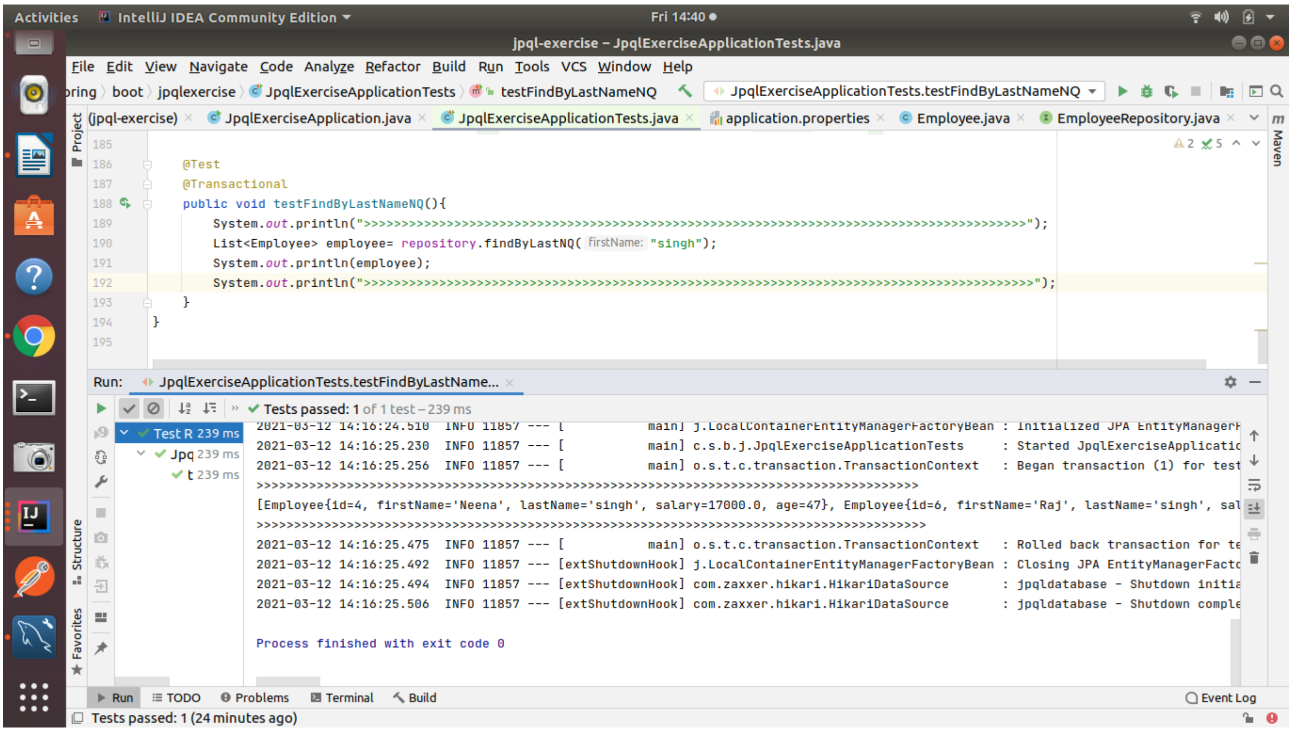
```java
//Q3 jpql
@Test
@Transactional
public void testDeleteEmployees(){
    repository.deleteEmployees();
}
```

Native SQL Query:

1. Display the id, first name, age of all employees where last name ends with "singh"



```java
//Q1 NQ
@Query(value = "select * from employee_table where emp_last_name=:lastName", nativeQuery = true)
List<Employee> findByLastNQ(@Param("lastName") String firstName);
```



2. Delete all employees with age greater than 45(Should be passed as a parameter)

```java
//Q2 NQ
@Modifying
@Query(value = "delete from employee_table where emp_age>:age", nativeQuery = true)
void deleteEmployeeHavingAgeGreaterThan(@Param("age") int age);
```

```java
//Q2 NQ
@Test
@Transactional
@Rollback(value = false)
public void testDeleteEmployeeHavingAgeGreaterThan(){

    repository.deleteEmployeeHavingAgeGreaterThan(45);
}
```

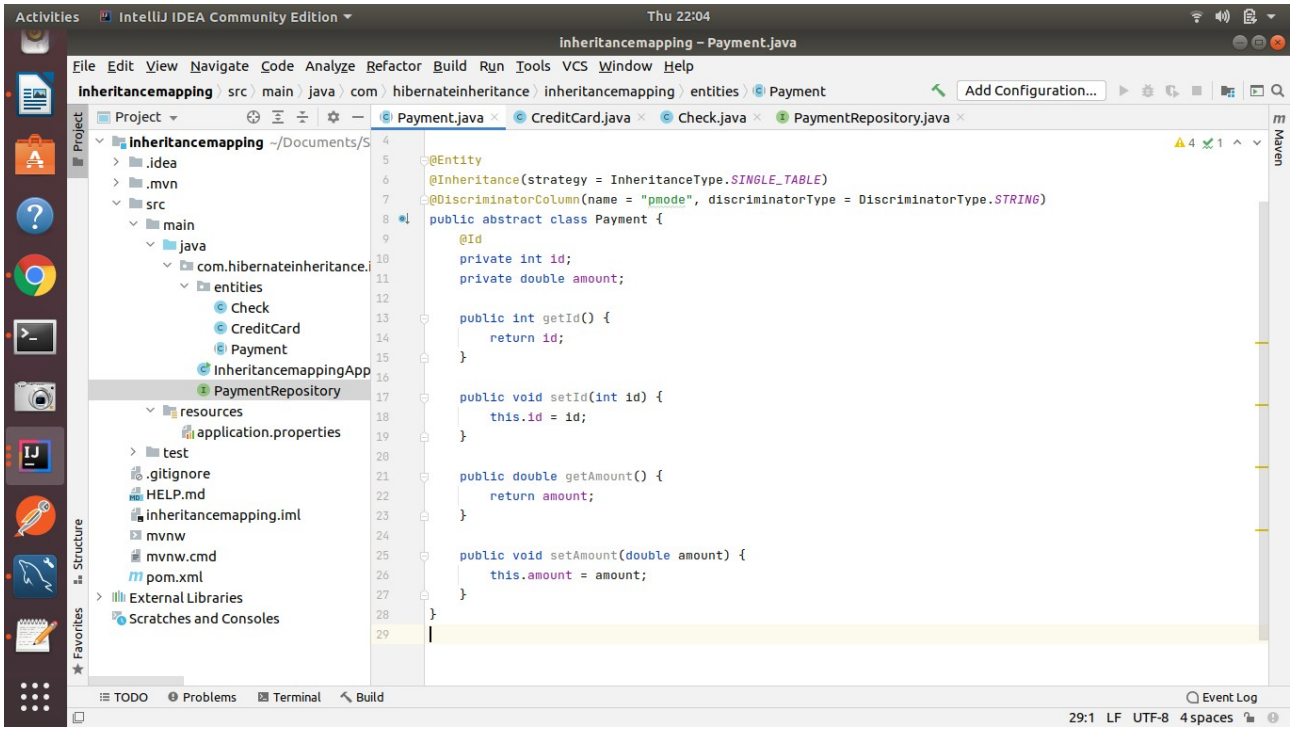
Result Grid | Filter Rows: | Edit: | Export/Import:

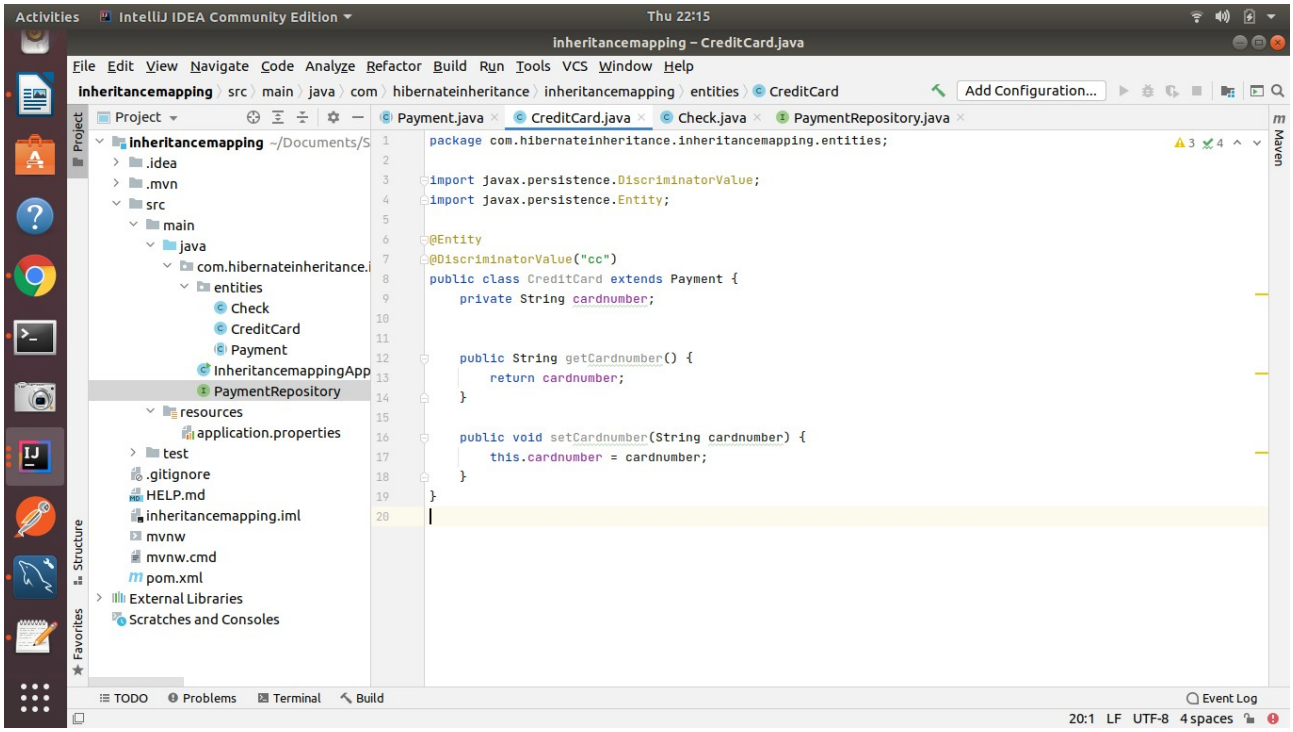
#	emp_id	emp_first_name	emp_last_name	emp_salary	emp_age
1	1	Naveen	Garg	20000.00	25
2	2	Bill	Gates	50000.00	35
3	5	Lex	De Hann	17000.00	23
4	6	Raj	singh	30000.00	28
5	7	Jitender	Kumar	70000.00	36
6	8	Ankit	Saini	45000.00	31
*	NULL	NULL	NULL	NULL	NULL

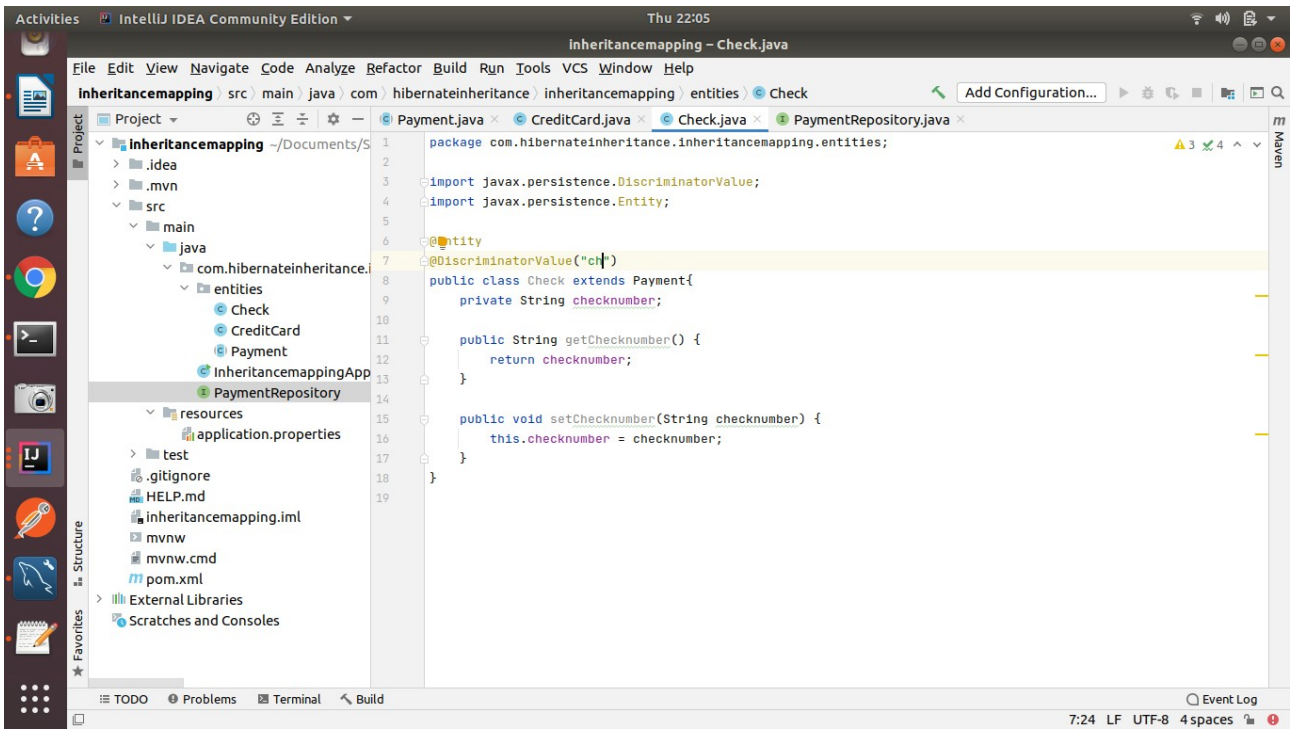
Inheritance Mapping:

1. Implement and demonstrate Single Table strategy.

```
create table payment(
id int PRIMARY KEY,
pmode varchar(2),
amount decimal(8,3) ,
cardnumber varchar(20),
checknumber varchar(20)
);
```







Payment.java | CreditCard.java | Check.java | PaymentRepository.java

```java
package com.hibernateinheritance.inheritancemapping;

import com.hibernateinheritance.inheritancemapping.entities.Payment;
import org.springframework.data.repository.CrudRepository;

public interface PaymentRepository extends CrudRepository<Payment, Integer> {
}
```

```java
@Test
public void createPayment(){
    CreditCard cc = new CreditCard();
    cc.setId(213);
    cc.setAmount(5000);
    cc.setCardnumber("9876543210");
    repository.save(cc);
}

@Test
public void createCheckPayment(){
    Check ch = new Check();
    ch.setId(214);
    ch.setAmount(6000);
    ch.setChecknumber("9876543210");
    repository.save(ch);
}
```

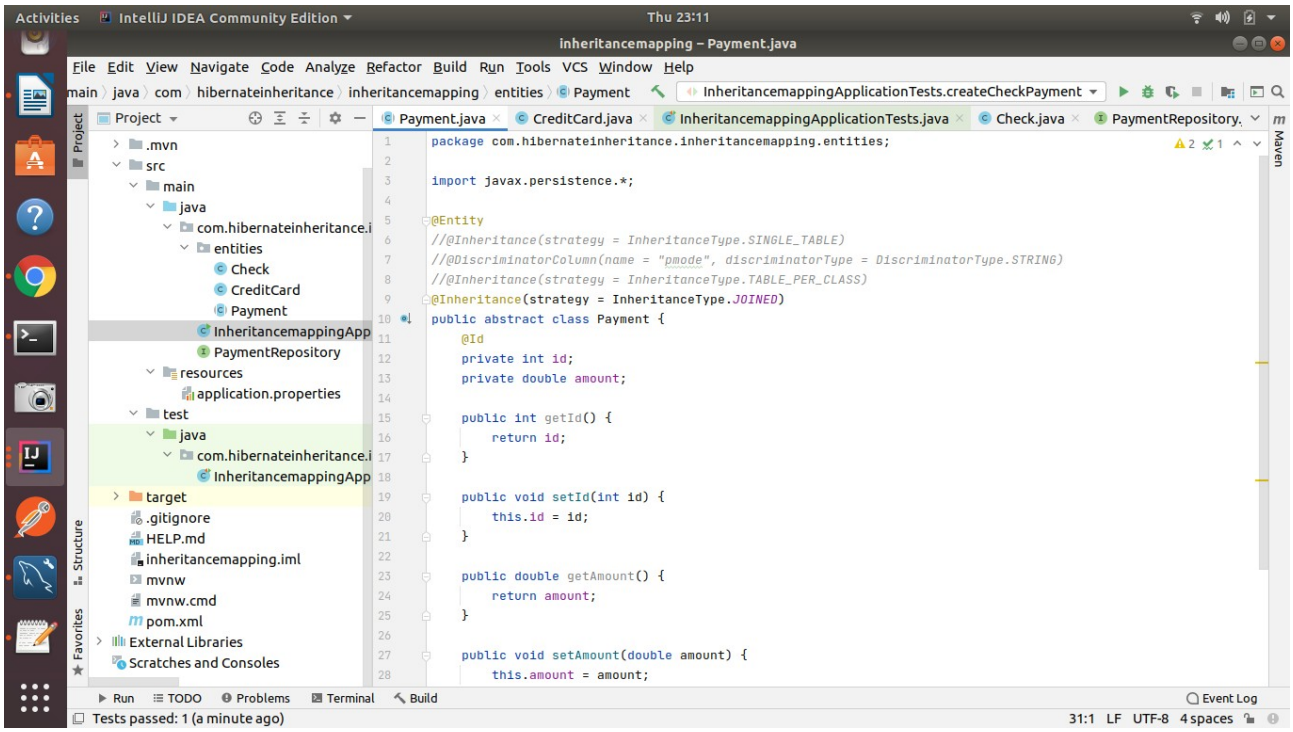
#	id	pmode	amount	cardnumber	checknumber
1	213	cc	5000.000	9876543210	NULL
2	214	ch	6000.000	NULL	9876543210
*	NULL	NULL	NULL	NULL	NULL

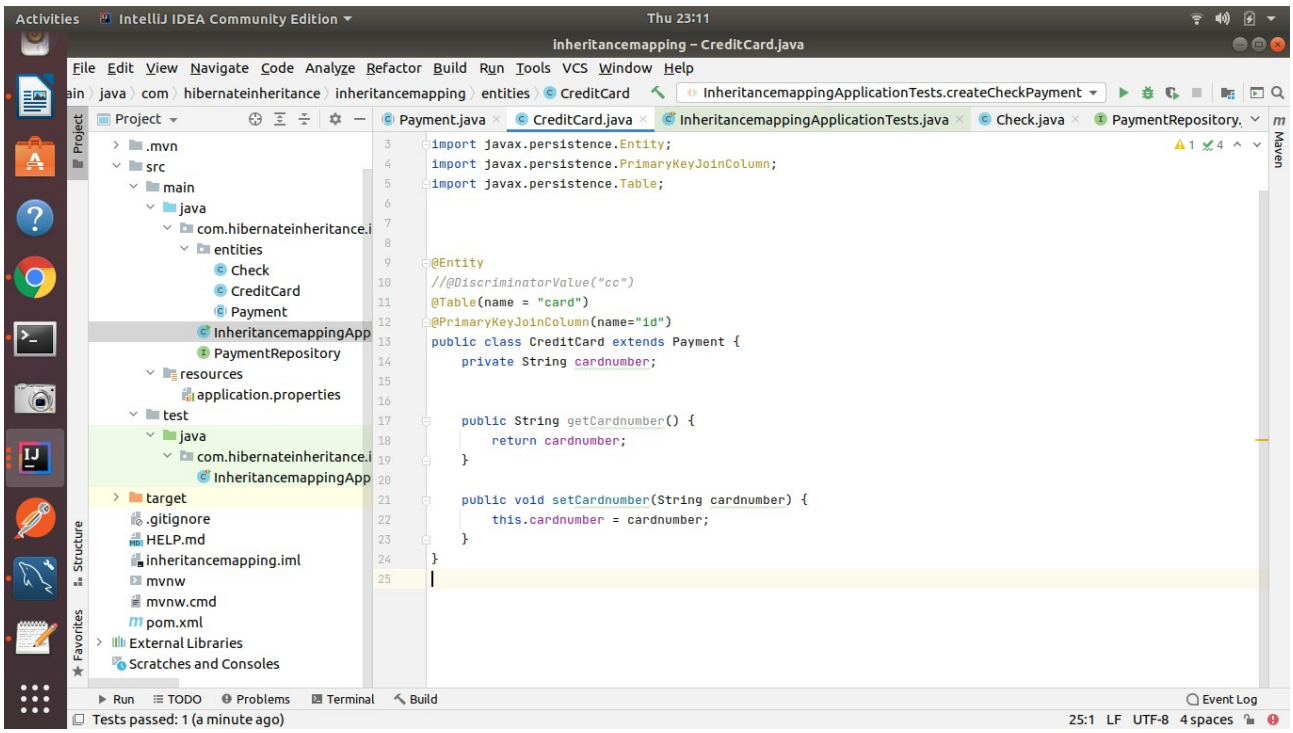
2. Implement and demonstrate Joined strategy.

```sql
create table payment(
id int PRIMARY KEY NOT NULL AUTO_INCREMENT,
amount decimal(8,3)
);

create table card(
id int ,
cardnumber varchar(20),
 FOREIGN KEY (id)
REFERENCES payment(id)
)

create table bankcheck(
id int ,
checknumber varchar(20),
FOREIGN KEY (id)
REFERENCES payment(id)
)
```






```java
@Test
public void createPayment(){
    CreditCard cc = new CreditCard();
    cc.setId(213);
    cc.setAmount(5000);
    cc.setCardnumber("9876543210");
    repository.save(cc);
}

@Test
public void createCheckPayment(){
    Check ch = new Check();
    ch.setId(214);
    ch.setAmount(6000);
    ch.setChecknumber("9876543210");
    repository.save(ch);
}
```

#	id	amount
1	213	5000.000
2	214	6000.000
*	NULL	NULL

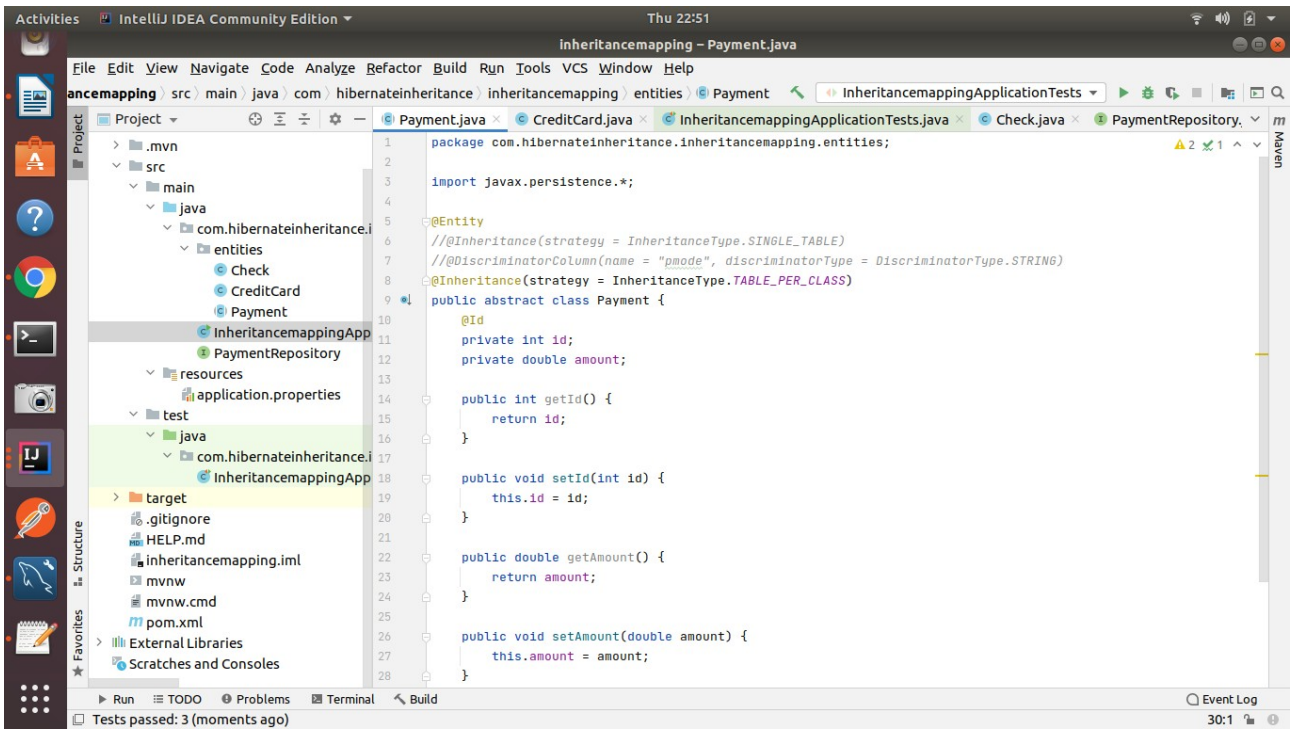
#	id	checknumber
1	214	9876543210

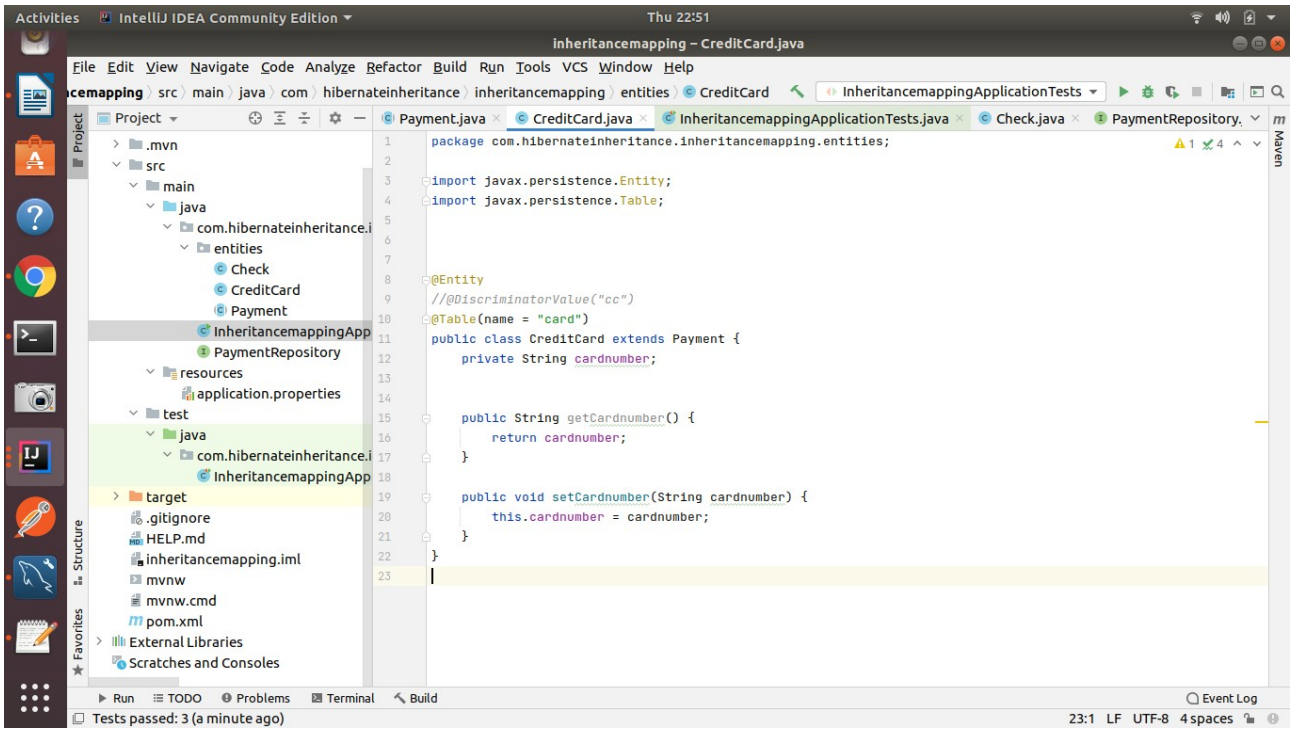
#	id	cardnumber
1	213	9876543210

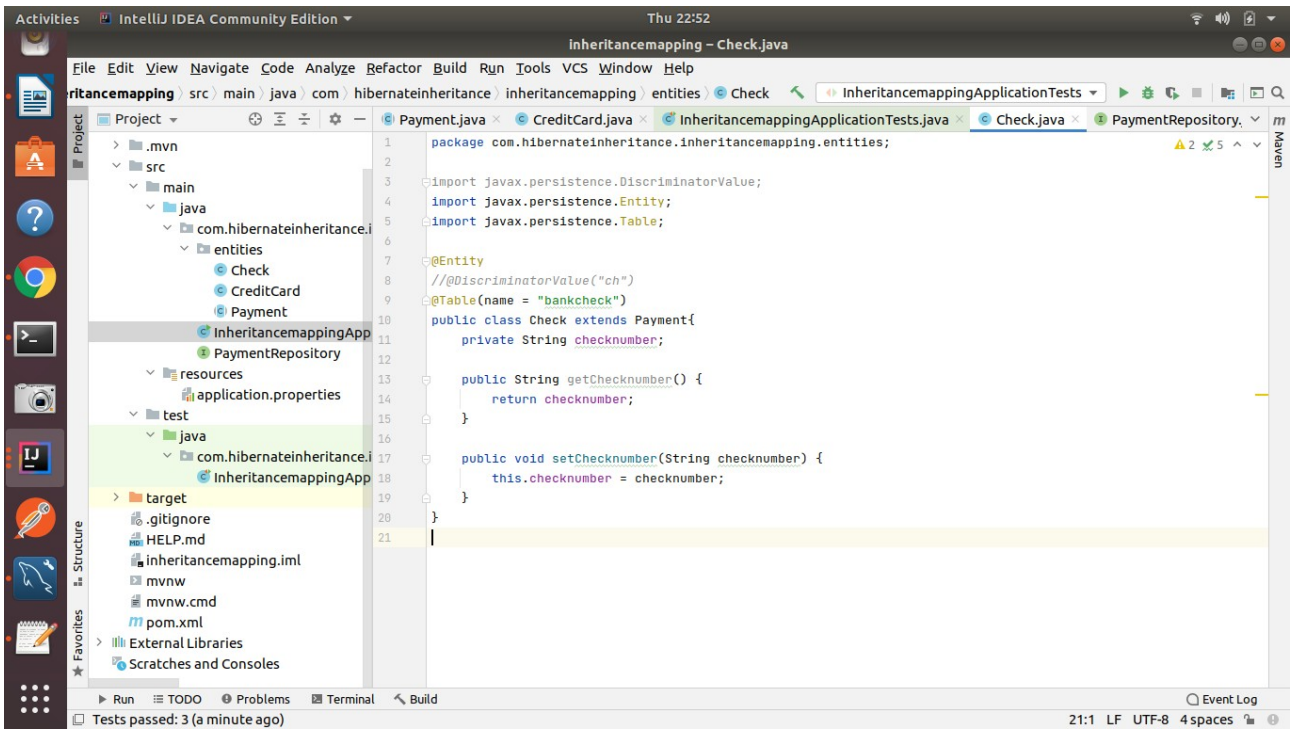
3. Implement and demonstrate Table Per Class strategy.

```
create table card(
id int PRIMARY KEY,
amount decimal(8,3),
cardnumber varchar(20)
)

create table bankcheck(
id int PRIMARY KEY,
amount decimal(8,3),
checknumber varchar(20)
)
```


Activities
IntelliJ IDEA Community Edition
Thu 22:51
inheritancemapping – Payment.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ancemapping › src › main › java › com › hibernateinheritance › inheritancemapping › entities › Payment
InheritancemappingApplicationTests
Project
Payment.java
CreditCard.java
InheritancemappingApplicationTests.java
Check.java
PaymentRepository.
.mvn
src
main
java
com.hibernateinheritance.i
entities
Check
CreditCard
Payment
InheritancemappingApp
PaymentRepository
resources
application.properties
test
java
com.hibernateinheritance.i
InheritancemappingApp
target
.gitignore
HELP.md
inheritancemapping.iml
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
package com.hibernateinheritance.inheritancemapping.entities;
import javax.persistence.*;
@Entity
//@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
//@DiscriminatorColumn(name = "pmode", discriminatorType = DiscriminatorType.STRING)
@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)
public abstract class Payment {
@Id
private int id;
private double amount;
public int getId() {
return id;
}
public void setId(int id) {
this.id = id;
}
public double getAmount() {
return amount;
}
public void setAmount(double amount) {
this.amount = amount;
}
Run
TODO
Problems
Terminal
Build
Event Log
Tests passed: 3 (moments ago)
30:1


Activities
IntelliJ IDEA Community Edition
Thu 22:51
inheritancemapping – CreditCard.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
cemapping › src › main › java › com › hibernateinheritance › inheritancemapping › entities › CreditCard
InheritancemappingApplicationTests
Project
Payment.java
CreditCard.java
InheritancemappingApplicationTests.java
Check.java
PaymentRepository.
.mvn
src
main
java
com.hibernateinheritance.i
entities
Check
CreditCard
Payment
InheritancemappingApp
PaymentRepository
resources
application.properties
test
java
com.hibernateinheritance.i
InheritancemappingApp
target
.gitignore
HELP.md
inheritancemapping.iml
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
package com.hibernateinheritance.inheritancemapping.entities;
import javax.persistence.Entity;
import javax.persistence.Table;
@Entity
//@DiscriminatorValue("cc")
@Table(name = "card")
public class CreditCard extends Payment {
private String cardnumber;
public String getCardnumber() {
return cardnumber;
}
public void setCardnumber(String cardnumber) {
this.cardnumber = cardnumber;
}
}
Run
TODO
Problems
Terminal
Build
Event Log
Tests passed: 3 (a minute ago)
23:1 LF UTF-8 4 spaces




```java
@Test
public void createPayment(){
    CreditCard cc = new CreditCard();
    cc.setId(213);
    cc.setAmount(5000);
    cc.setCardnumber("9876543210");
    repository.save(cc);
}

@Test
public void createCheckPayment(){
    Check ch = new Check();
    ch.setId(214);
    ch.setAmount(6000);
    ch.setChecknumber("9876543210");
    repository.save(ch);
}

}
```

#	id	amount	checknumber
1	214	6000.000	9876543210
*	NULL	NULL	NULL

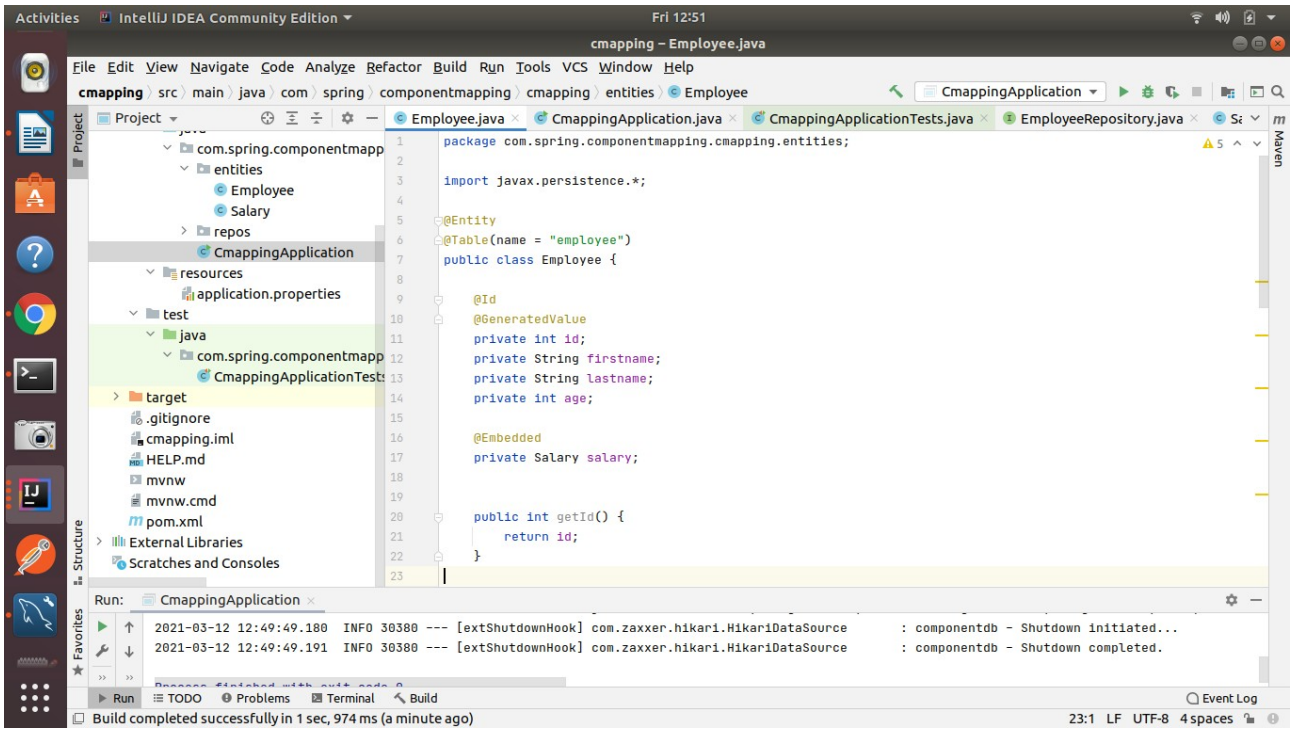
#	id	amount	cardnumber
1	213	5000.000	9876543210
*	NULL	NULL	NULL

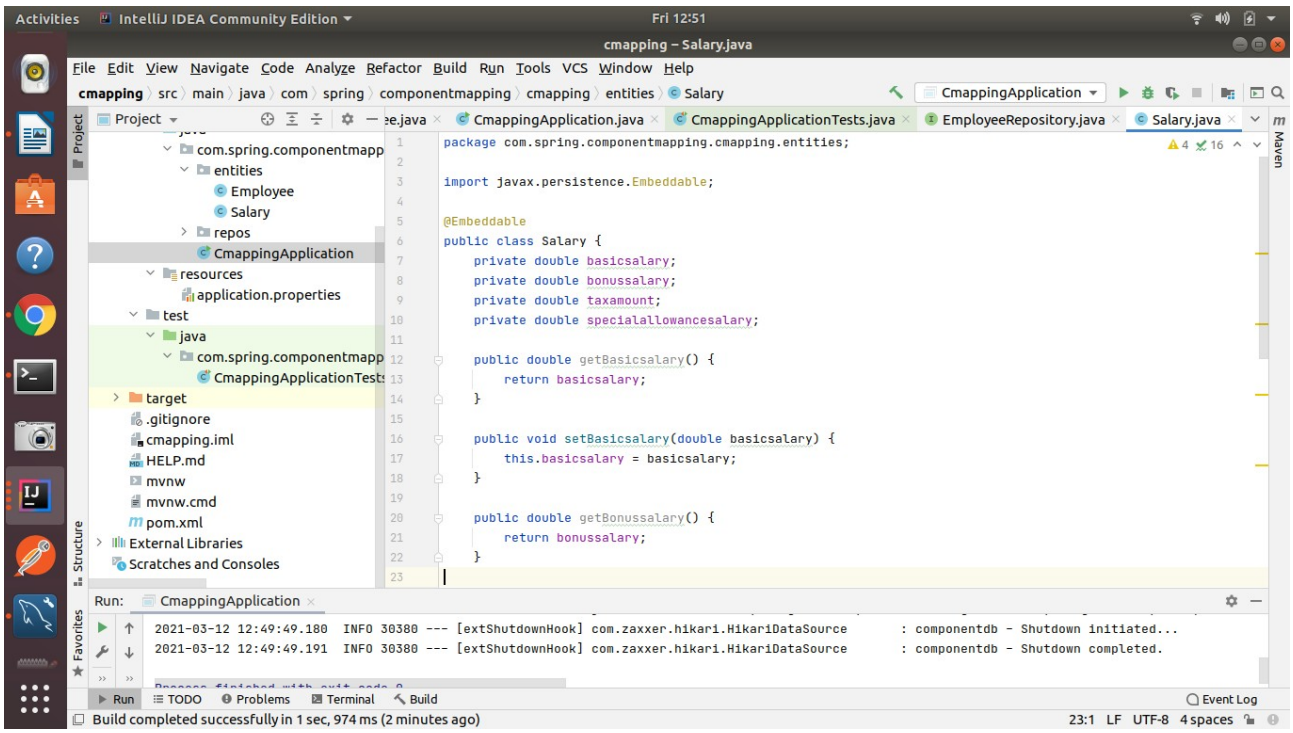
Component Mapping:

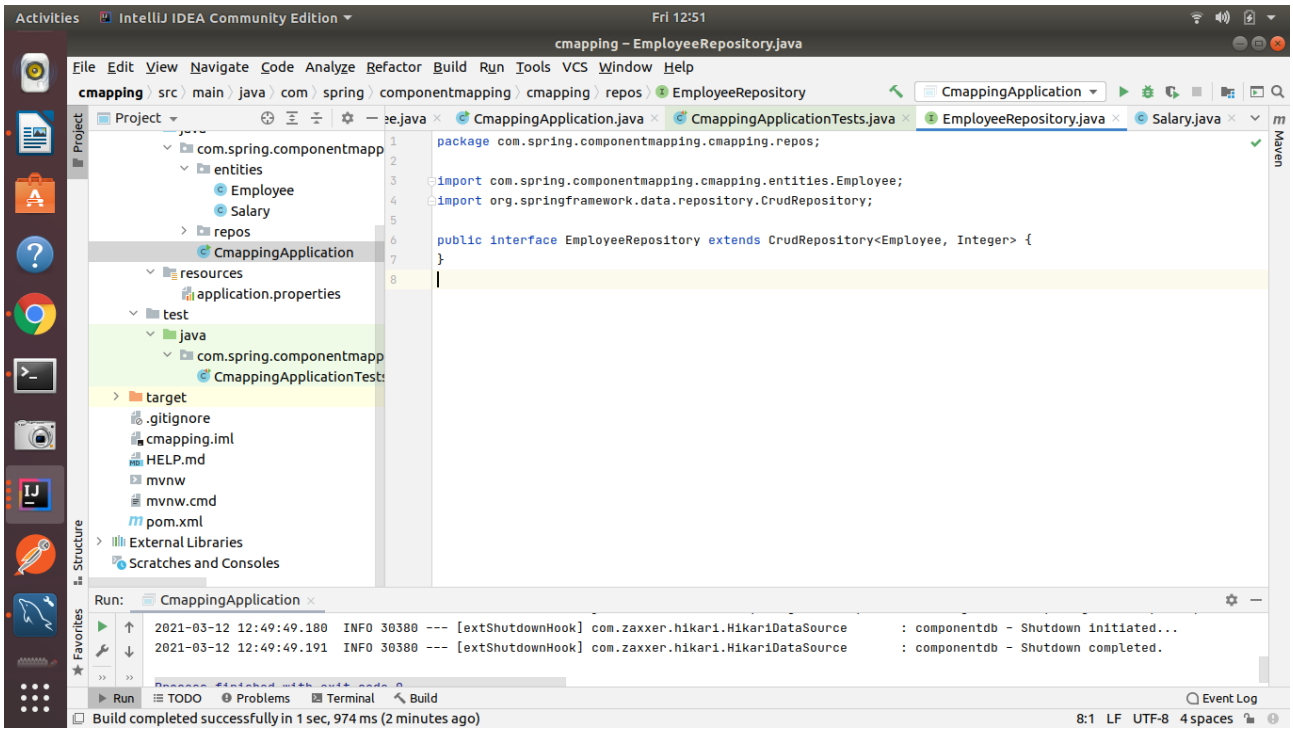
1. Implement and demonstrate Embedded mapping using employee table having following fields: id, firstName, lastName, age, basicSalary, bonusSalary, taxAmount, specialAllowanceSalary.

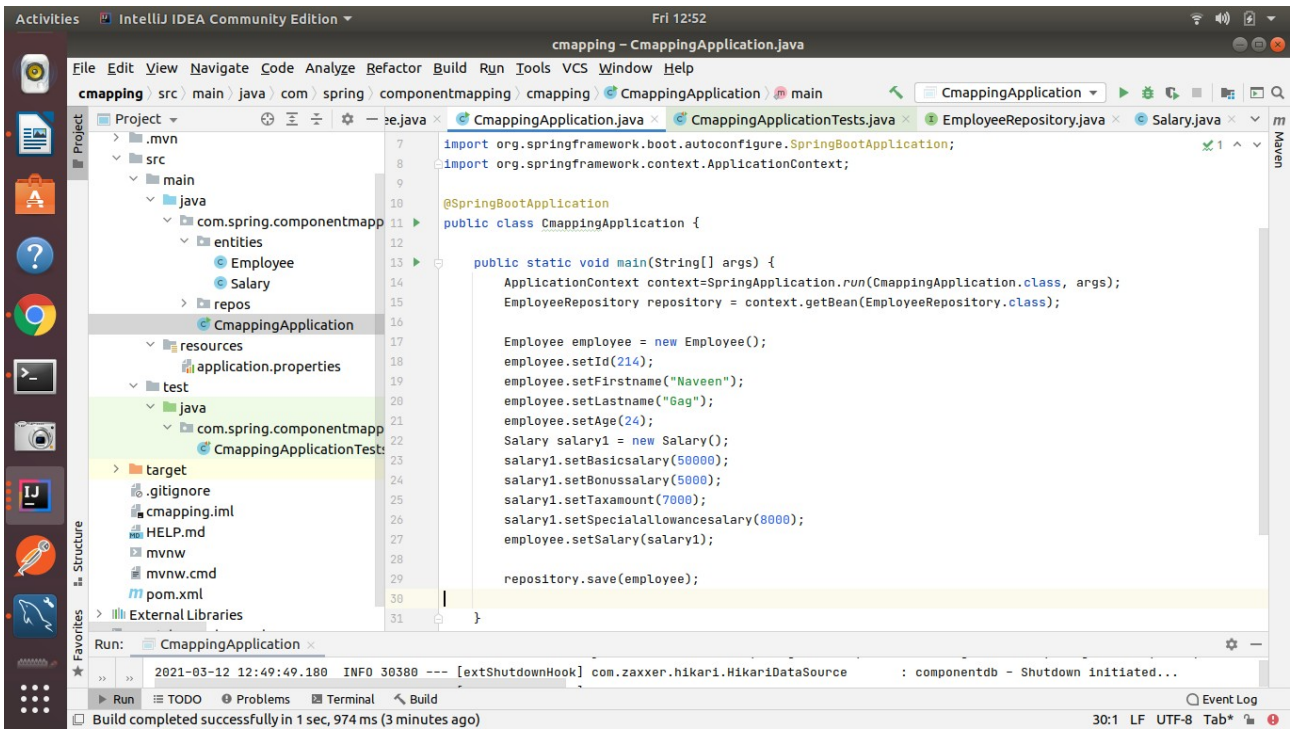
```
create table employee(
id int,
firstname varchar(20),
lastname varchar(20),
age int,
basicsalary decimal(8,3),
bonussalary decimal(8,3),
taxamount decimal(8,3),
specialallowancesalary decimal(8,3)
)

select * from employee
```




Activities
IntelliJ IDEA Community Edition
Fri 12:51
cmapping – Salary.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
cmapping › src › main › java › com › spring › componentmapping › cmapping › entities › Salary
CmappingApplication
Project
CmappingApplication.java
CmappingApplicationTests.java
EmployeeRepository.java
Salary.java
com.spring.componentmapp
entities
Employee
Salary
repos
CmappingApplication
resources
application.properties
test
java
com.spring.componentmapp
CmappingApplicationTest
target
.gitignore
cmapping.iml
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
package com.spring.componentmapping.cmapping.entities;
import javax.persistence.Embeddable;
@Embeddable
public class Salary {
private double basicsalary;
private double bonussalary;
private double taxamount;
private double specialallowancesalary;
public double getBasicsalary() {
return basicsalary;
}
public void setBasicsalary(double basicsalary) {
this.basicsalary = basicsalary;
}
public double getBonussalary() {
return bonussalary;
}
Run: CmappingApplication
2021-03-12 12:49:49.180 INFO 30380 --- [extShutdownHook] com.zaxxer.hikari.HikariDataSource : componentdb - Shutdown initiated...
2021-03-12 12:49:49.191 INFO 30380 --- [extShutdownHook] com.zaxxer.hikari.HikariDataSource : componentdb - Shutdown completed.
Run
TODO
Problems
Terminal
Build
Event Log
Build completed successfully in 1 sec, 974 ms (2 minutes ago)
23:1 LF UTF-8 4 spaces


Activities
IntelliJ IDEA Community Edition
Fri 12:51
cmapping – EmployeeRepository.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
cmapping › src › main › java › com › spring › componentmapping › cmapping › repos › EmployeeRepository
CmappingApplication
Project
CmappingApplication.java
CmappingApplicationTests.java
EmployeeRepository.java
Salary.java
com.spring.componentmapp
entities
Employee
Salary
repos
CmappingApplication
resources
application.properties
test
java
com.spring.componentmapp
CmappingApplicationTest
target
.gitignore
cmapping.iml
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles
package com.spring.componentmapping.cmapping.repos;
import com.spring.componentmapping.cmapping.entities.Employee;
import org.springframework.data.repository.CrudRepository;
public interface EmployeeRepository extends CrudRepository<Employee, Integer> {
}
Run: CmappingApplication
2021-03-12 12:49:49.180 INFO 30380 --- [extShutdownHook] com.zaxxer.hikari.HikariDataSource : componentdb - Shutdown initiated...
2021-03-12 12:49:49.191 INFO 30380 --- [extShutdownHook] com.zaxxer.hikari.HikariDataSource : componentdb - Shutdown completed.
Run
TODO
Problems
Terminal
Build
Event Log
Build completed successfully in 1 sec, 974 ms (2 minutes ago)
8:1 LF UTF-8 4 spaces



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

#	id	firstname	lastname	age	basicsalary	bonussalary	taxamount	specialallowancesalary
1	5	Naveen	Gag	24	50000.000	5000.000	7000.000	8000.000