LAB 10

Q.1. Develop a basic Create, read operation using Hibernate for a simple entity, such as Student.

Program files:

a). pom.xml

Program:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.demo</groupId>
  <artifactId>college_management_system</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>college_management_system</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
<dependency>
<groupId>org.hibernate</groupId>
<artifactId>hibernate-core</artifactId>
<version>5.6.15.Final</version>
</dependency>
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>8.0.30</version>
</dependency>
     <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

b). hibernateUtil.java

program:

```java
package com.demo.college_management_system;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
public class HibernateUtil {
private static final SessionFactory = buildSessionFactory();
private static SessionFactory buildSessionFactory() {
try {
return new
Configuration().configure("hibernate.cfg.xml").addAnnotatedClass(Student.class)
.buildSessionFactory();
} catch (Throwable ex) {
throw new ExceptionInInitializerError(ex);
}
}
public static SessionFactory getSessionFactory() {
return sessionFactory;
}
}
```

c). hibernate.cfg.xml

program:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<!-- Database connection settings -->
<property name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
<property
name="connection.url">jdbc:mysql://localhost:3306/college_management_system</prop
erty>
<property name="connection.username">root</property>
<property name="connection.password">Abbu@05</property>
<!-- Automatically create or update database schema -->
<property name="hbm2ddl.auto">update</property>
<!-- Specify the dialect for your database -->
<property name="dialect">org.hibernate.dialect.MySQL8Dialect</property>
<!-- Enable or disable showing SQL statements in the console -->
<property name="show_sql">true</property>
<property name="format_sql">true</property>
<!-- Specify the package(s) where your entity classes are located →
<mapping class="com.sms.Student"/>
<! Additional mappings can be included here -->
</session-factory>
</hibernate-configuration>
```

d). Student.java

Program:

```
package com.demo.college_management_system;
import java.time.LocalDate;
import java.time.LocalDate;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
@Entity
public class Student {
@Id
@Column(name = "StudentId", length = 10)
private String studentId;
@Column(name = "FirstName", length = 50)
private String firstName;
@Column(name = "LastName", length = 25)
private String lastName;
@Column(name = "DateOfBirth")
private LocalDate dateOfBirth;
@Column(name = "Gender", length = 25)
private String gender;
@Column(name = "Email", length = 30)
private String email;
@Column(name = "Phone", length = 25)
private String phone;
//Setter and Getter
public String getStudentId() {
return studentId;
}
public void setStudentId(String studentId) {
this.studentId = studentId;
}
public String getFirstName() {
return firstName;
}
public void setFirstName(String firstName) {
this.firstName = firstName;
}
public String getLastName() {
```

```
return lastName;
}
public void setLastName(String lastName) {
this.lastName = lastName;
}
public LocalDate getDateOfBirth() {
return dateOfBirth;
}
public void setDateOfBirth(LocalDate dateOfBirth) {
this.dateOfBirth = dateOfBirth;
}
public String getGender() {
return gender;
}
public void setGender(String gender) {
this.gender = gender;
}
public String getEmail() {
return email;
}
public void setEmail(String email) {
this.email = email;
}
public String getPhone() {
return phone;
}
public void setPhone(String phone) {
this.phone = phone;
}
//All argument Constructor
public Student(String studentId, String firstName, String
lastName, LocalDate dateOfBirth, String gender,String email,
String phone) {
super();
this.studentId = studentId;
this.firstName = firstName;
this.lastName = lastName;
this.dateOfBirth = dateOfBirth;
this.gender = gender;
this.email = email;
this.phone = phone;
}
```

```
//Default Constructor
public Student() {
super();
}
//ToString method
@Override
public String toString() {
return "Student [studentId=" + studentId + ", firstName=" +
firstName + ",lastName=" + lastName+ ", dateOfBirth=" +
dateOfBirth + ", gender=" + gender + ",email=" + email + ",
phone=" + phone + "]";
}
}
```

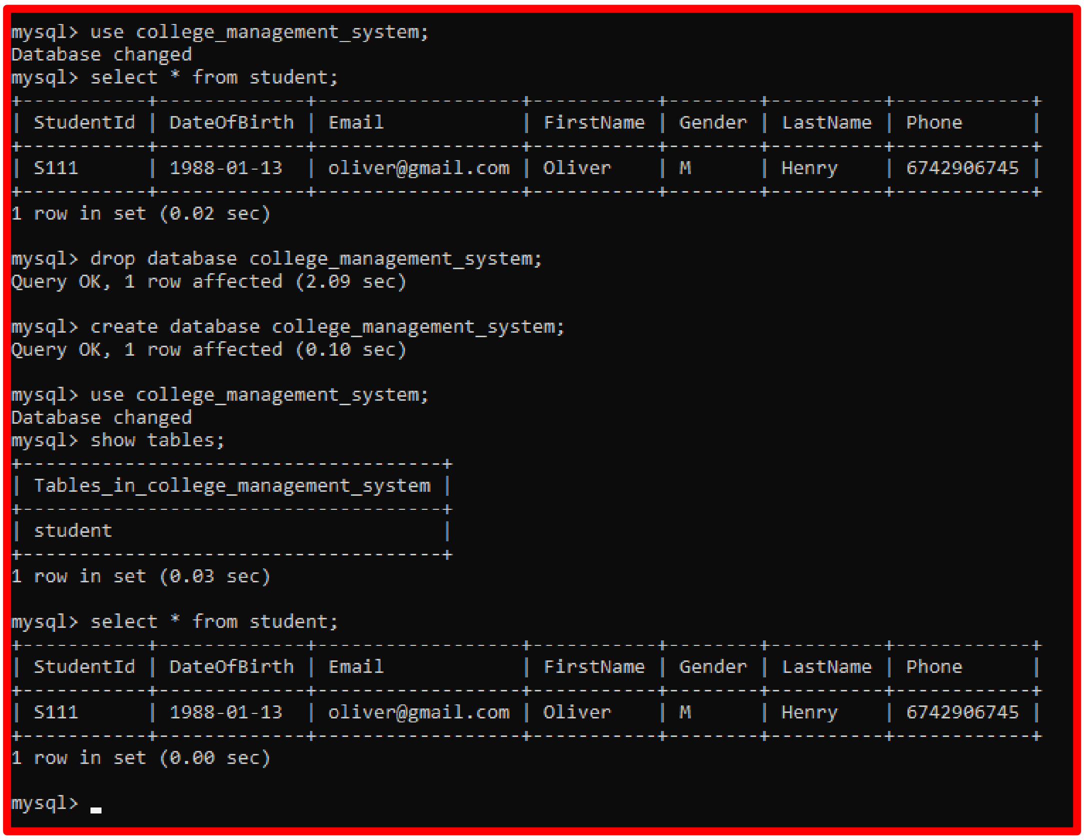
e). App.java

Program:

```
package com.demo.college_management_system;
import java.time.LocalDate;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
public class App {
public static void main(String[] args) {
// Obtain a Hibernate SessionFactory
SessionFactory factory = HibernateUtil.getSessionFactory();
// Create a new Student
LocalDate date1 = LocalDate.of(1988, 1, 13);
Student Student1 = new Student("S111", "Oliver", "Henry",
date1, "M","oliver@gmail.com", "6742906745");
// Open a new session
Session session = factory.openSession();
// Begin a transaction
Transaction transaction = session.beginTransaction();
// Save the student to the database
session.save(Student1);
// Commit the transaction
transaction.commit();
// Close the Session
session.close();
// Close the Session Factory
factory.close();
}
}
```

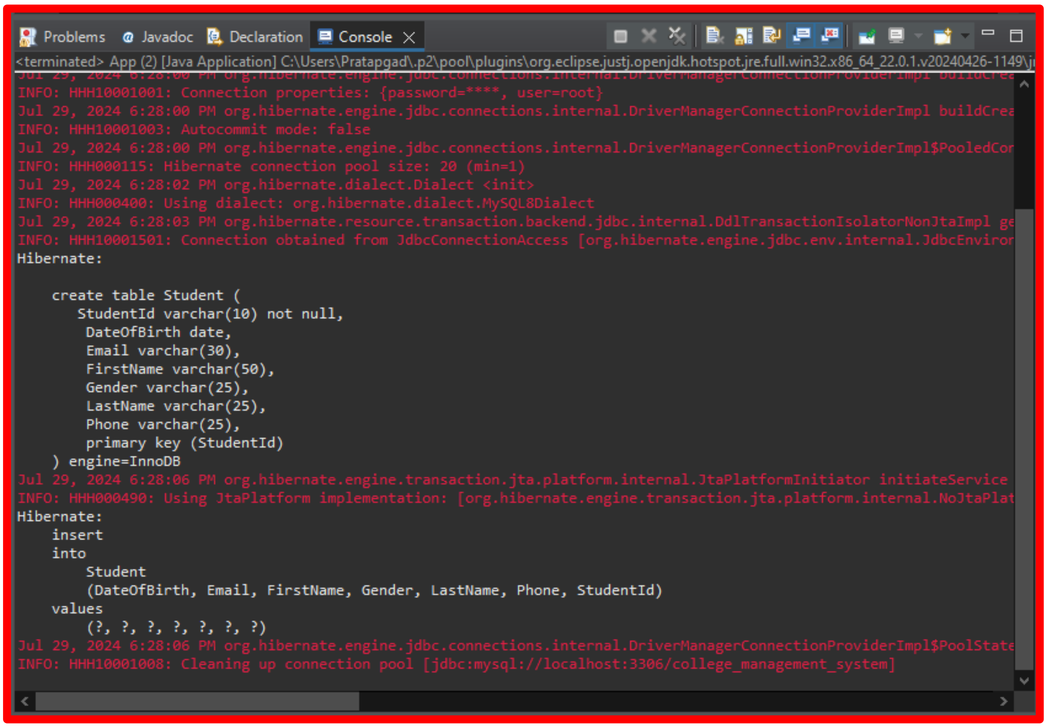
Output Screen:

Database Screen:



```
mysql> use college_management_system;
Database changed
mysql> select * from student;
+-----------+-------------+------------------+-----------+--------+----------+------------+
| StudentId | DateOfBirth | Email            | FirstName | Gender | LastName | Phone      |
+-----------+-------------+------------------+-----------+--------+----------+------------+
| S111      | 1988-01-13  | oliver@gmail.com | Oliver    | M      | Henry    | 6742906745 |
+-----------+-------------+------------------+-----------+--------+----------+------------+
1 row in set (0.02 sec)

mysql> drop database college_management_system;
Query OK, 1 row affected (2.09 sec)

mysql> create database college_management_system;
Query OK, 1 row affected (0.10 sec)

mysql> use college_management_system;
Database changed
mysql> show tables;
+-------------------------------------+
| Tables_in_college_management_system |
+-------------------------------------+
| student                             |
+-------------------------------------+
1 row in set (0.03 sec)

mysql> select * from student;
+-----------+-------------+------------------+-----------+--------+----------+------------+
| StudentId | DateOfBirth | Email            | FirstName | Gender | LastName | Phone      |
+-----------+-------------+------------------+-----------+--------+----------+------------+
| S111      | 1988-01-13  | oliver@gmail.com | Oliver    | M      | Henry    | 6742906745 |
+-----------+-------------+------------------+-----------+--------+----------+------------+
1 row in set (0.00 sec)

mysql> _
```

Eclipse Screen:

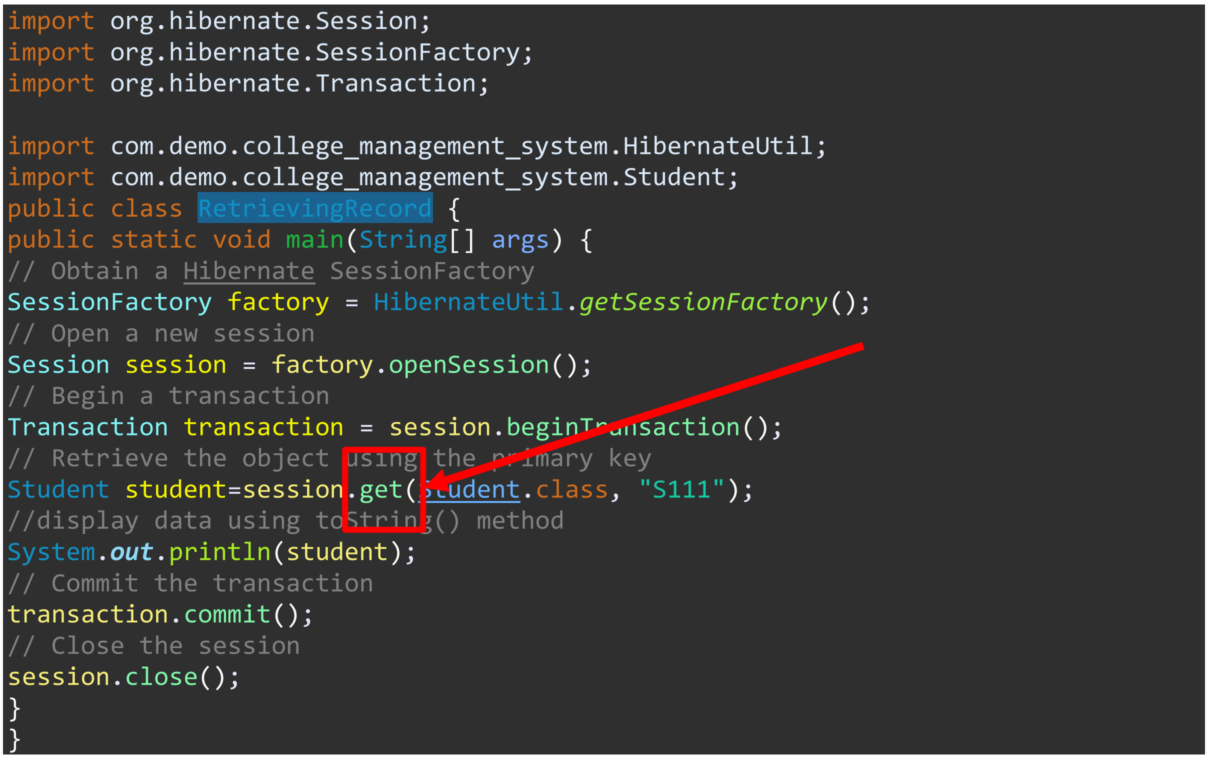


```
<terminated> App (2) [Java Application] C:\Users\Pratapgad\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.1.v20240426-1149\j
Jul 29, 2024 6:28:00 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCrea
INFO: HHH10001001: Connection properties: {password=****, user=root}
Jul 29, 2024 6:28:00 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCrea
INFO: HHH10001003: Autocommit mode: false
Jul 29, 2024 6:28:00 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledCor
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Jul 29, 2024 6:28:02 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
Jul 29, 2024 6:28:03 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl ge
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnviror
Hibernate: 

    create table Student (
       StudentId varchar(10) not null,
        DateOfBirth date,
        Email varchar(30),
        FirstName varchar(50),
        Gender varchar(25),
        LastName varchar(25),
        Phone varchar(25),
        primary key (StudentId)
    ) engine=InnoDB
Jul 29, 2024 6:28:06 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlat
Hibernate: 
    insert 
    into
        Student
        (DateOfBirth, Email, FirstName, Gender, LastName, Phone, StudentId) 
    values
        (?, ?, ?, ?, ?, ?, ?)
Jul 29, 2024 6:28:06 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost:3306/college_management_system]
```

Q.2. Use get() method to fetch a student object with an ID that doesn't exist in the database. What will be the result, and how would you handle it?

Program Files:

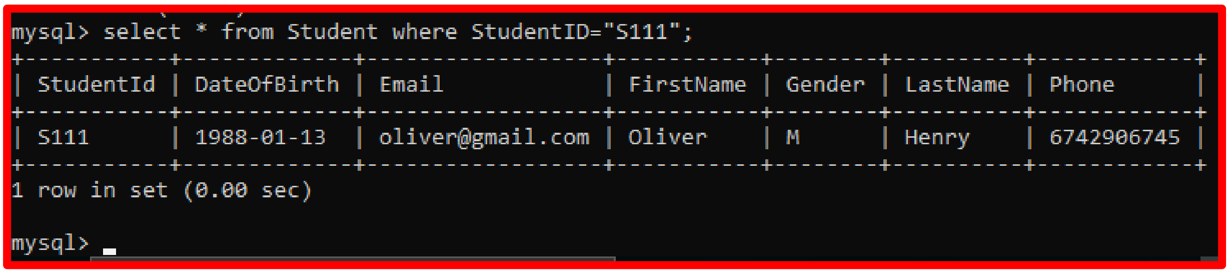
RetrieveRecord.java



```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;

import com.demo.college_management_system.HibernateUtil;
import com.demo.college_management_system.Student;
public class RetrievingRecord {
public static void main(String[] args) {
// Obtain a Hibernate SessionFactory
SessionFactory factory = HibernateUtil.getSessionFactory();
// Open a new session
Session session = factory.openSession();
// Begin a transaction
Transaction transaction = session.beginTransaction();
// Retrieve the object using the primary key
Student student=session.get(Student.class, "S111");
//display data using toString() method
System.out.println(student);
// Commit the transaction
transaction.commit();
// Close the session
session.close();
}
}
```

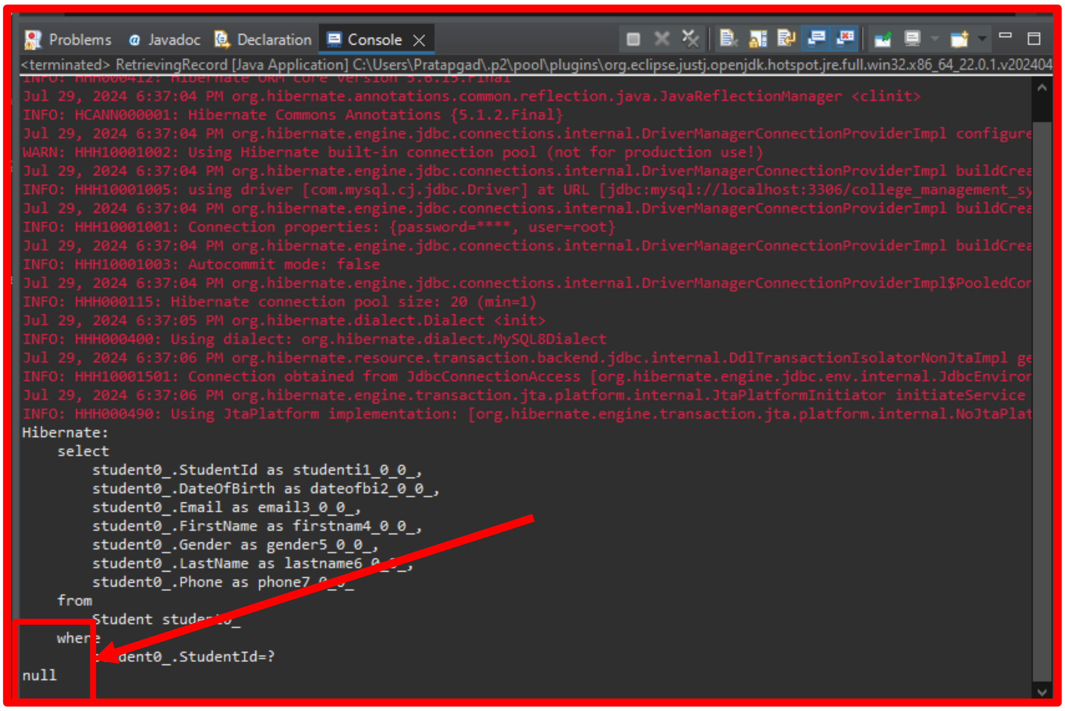
Output Screen:

Database:



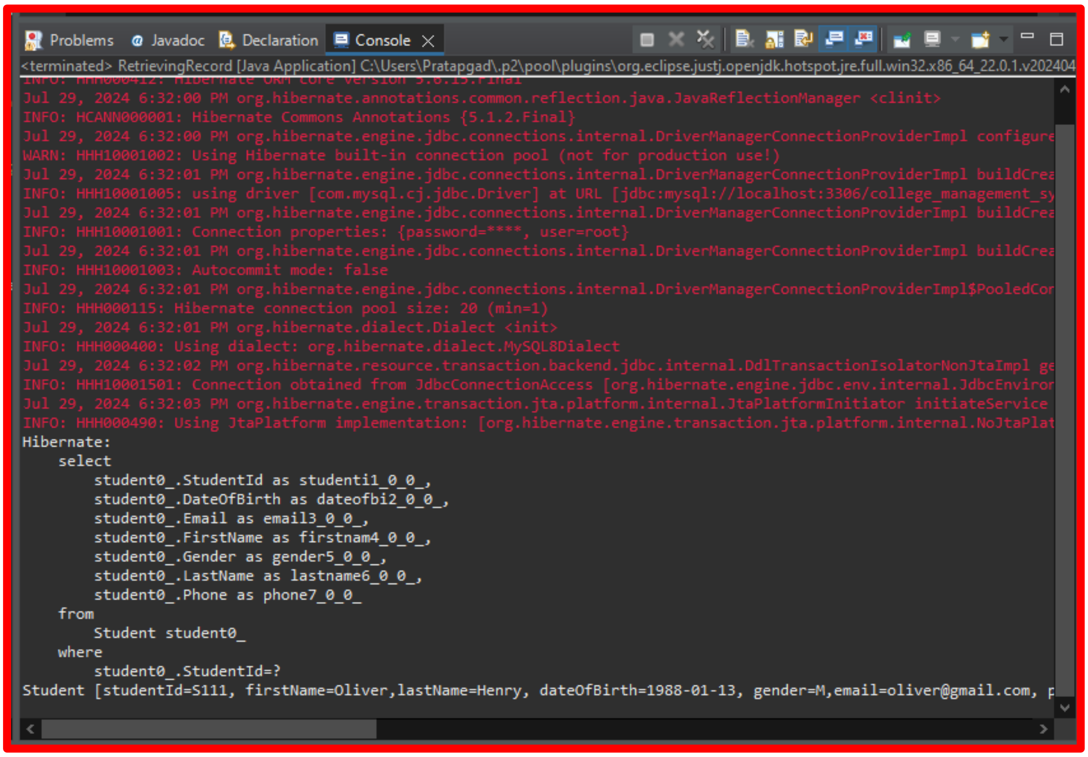
```
mysql> select * from Student where StudentID="S111";
+-----------+-------------+-------------------+-----------+--------+----------+------------+
| StudentId | DateOfBirth | Email             | FirstName | Gender | LastName | Phone      |
+-----------+-------------+-------------------+-----------+--------+----------+------------+
| S111      | 1988-01-13  | oliver@gmail.com  | Oliver    | M      | Henry    | 6742906745 |
+-----------+-------------+-------------------+-----------+--------+----------+------------+
1 row in set (0.00 sec)

mysql> _
```

<u>Eclipse:</u> If the parameter **(StudentID="S112")** of Object is not match with database's record then eclipse shows null:



```
<terminated> RetrievingRecord [Java Application] C:\Users\Pratapgad\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.1.v202404
INFO: HHH000412: Hibernate ORM core version 5.6.15.Final
Jul 29, 2024 6:37:04 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
Jul 29, 2024 6:37:04 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)
Jul 29, 2024 6:37:04 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCrea
INFO: HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/college_management_sy
Jul 29, 2024 6:37:04 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCrea
INFO: HHH10001001: Connection properties: {password=****, user=root}
Jul 29, 2024 6:37:04 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCrea
INFO: HHH10001003: Autocommit mode: false
Jul 29, 2024 6:37:04 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledCor
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Jul 29, 2024 6:37:05 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
Jul 29, 2024 6:37:06 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl ge
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnviror
Jul 29, 2024 6:37:06 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlat
Hibernate: 
    select
        student0_.StudentId as studenti1_0_0_,
        student0_.DateOfBirth as dateofbi2_0_0_,
        student0_.Email as email3_0_0_,
        student0_.FirstName as firstnam4_0_0_,
        student0_.Gender as gender5_0_0_,
        student0_.LastName as lastname6_0_0_,
        student0_.Phone as phone7_0_0_ 
    from
        Student student0_ 
    where
        student0_.StudentId=?
null
```

<u>Eclipse:</u> If the parameter **(StudentID="S111")** of Object is match with database's record then eclipse ran the program successfully and shows the record:

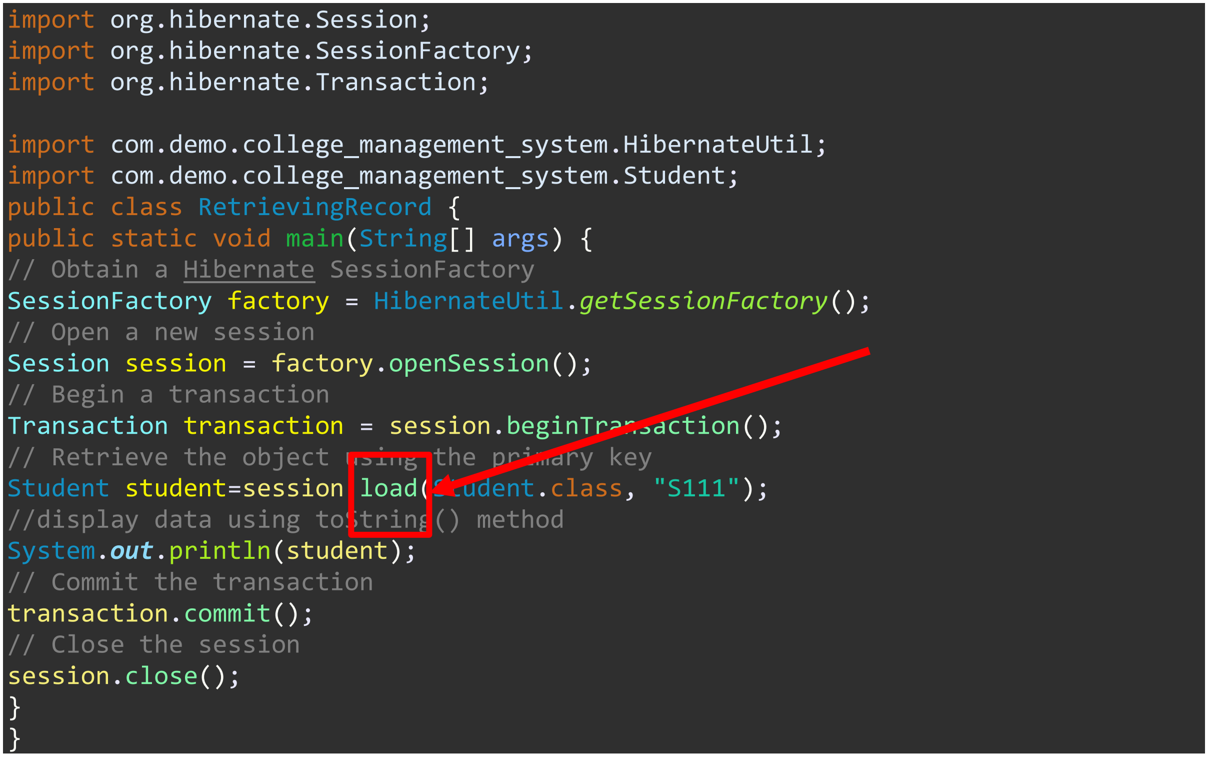


```
<terminated> RetrievingRecord [Java Application] C:\Users\Pratapgad\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.1.v202404
INFO: HHH000412: Hibernate ORM core version 5.6.15.Final
Jul 29, 2024 6:32:00 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
Jul 29, 2024 6:32:00 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)
Jul 29, 2024 6:32:01 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCrea
INFO: HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/college_management_sy
Jul 29, 2024 6:32:01 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCrea
INFO: HHH10001001: Connection properties: {password=****, user=root}
Jul 29, 2024 6:32:01 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCrea
INFO: HHH10001003: Autocommit mode: false
Jul 29, 2024 6:32:01 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledCor
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Jul 29, 2024 6:32:01 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
Jul 29, 2024 6:32:02 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl ge
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnviror
Jul 29, 2024 6:32:03 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlat
Hibernate: 
    select
        student0_.StudentId as studenti1_0_0_,
        student0_.DateOfBirth as dateofbi2_0_0_,
        student0_.Email as email3_0_0_,
        student0_.FirstName as firstnam4_0_0_,
        student0_.Gender as gender5_0_0_,
        student0_.LastName as lastname6_0_0_,
        student0_.Phone as phone7_0_0_ 
    from
        Student student0_ 
    where
        student0_.StudentId=?
Student [studentId=S111, firstName=Oliver,lastName=Henry, dateOfBirth=1988-01-13, gender=M,email=oliver@gmail.com, p
```

Q.3. Also demonstrate use of load() method.

Program Files:

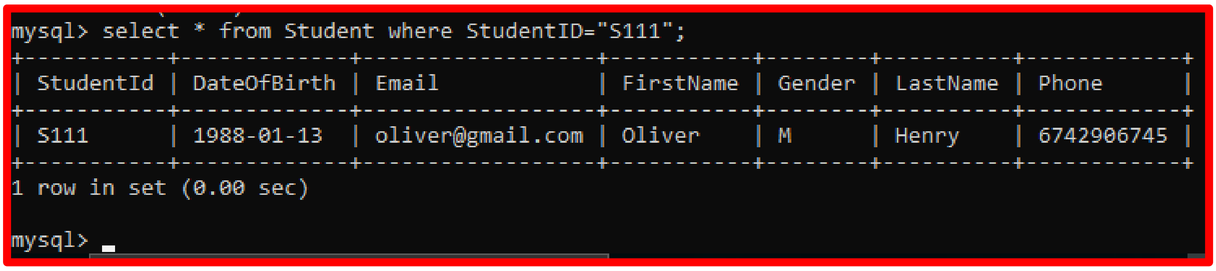
RetrieveRecord.java



```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;

import com.demo.college_management_system.HibernateUtil;
import com.demo.college_management_system.Student;
public class RetrievingRecord {
public static void main(String[] args) {
// Obtain a Hibernate SessionFactory
SessionFactory factory = HibernateUtil.getSessionFactory();
// Open a new session
Session session = factory.openSession();
// Begin a transaction
Transaction transaction = session.beginTransaction();
// Retrieve the object using the primary key
Student student=session.load(Student.class, "S111");
//display data using toString() method
System.out.println(student);
// Commit the transaction
transaction.commit();
// Close the session
session.close();
}
}
```

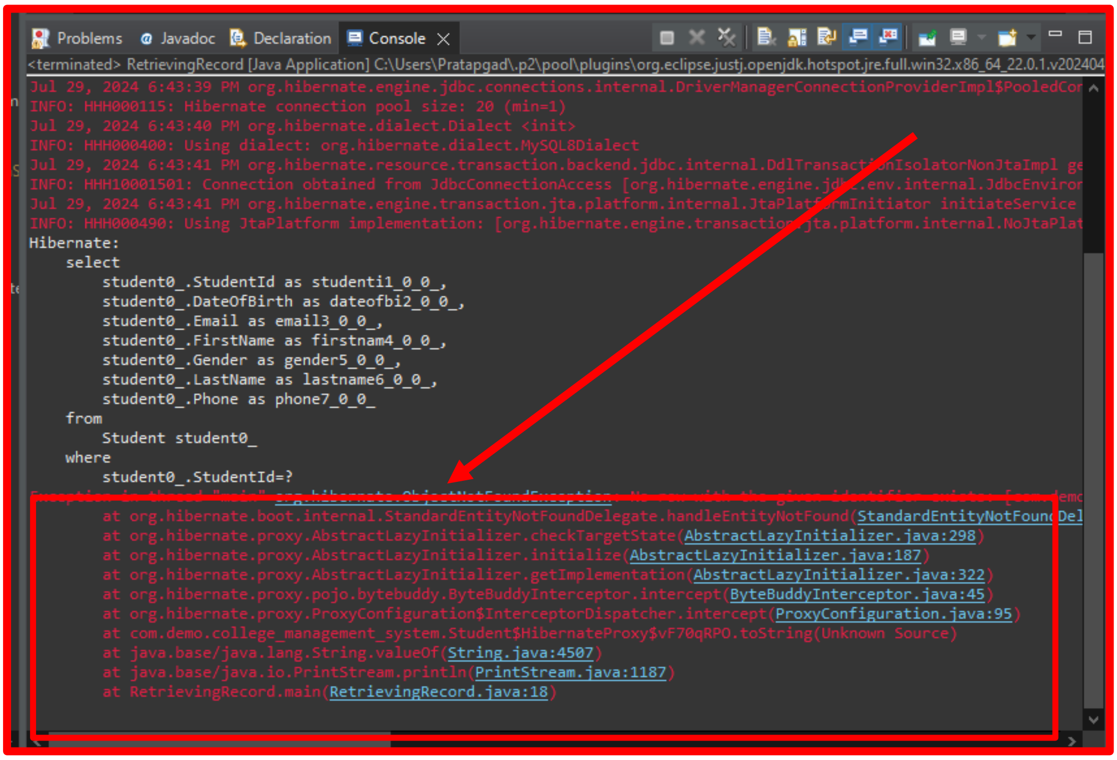
Output Screen:

Database:



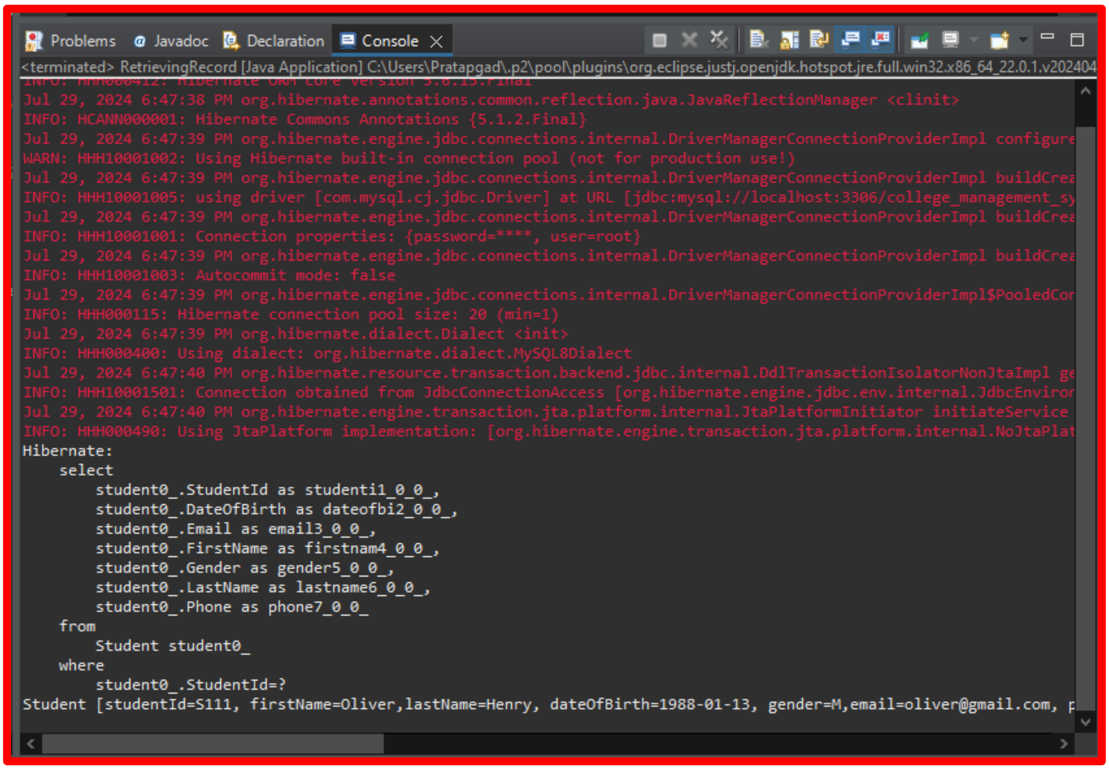
```
mysql> select * from Student where StudentID="S111";
+-----------+-------------+-------------------+-----------+--------+----------+------------+
| StudentId | DateOfBirth | Email             | FirstName | Gender | LastName | Phone      |
+-----------+-------------+-------------------+-----------+--------+----------+------------+
| S111      | 1988-01-13  | oliver@gmail.com  | Oliver    | M      | Henry    | 6742906745 |
+-----------+-------------+-------------------+-----------+--------+----------+------------+
1 row in set (0.00 sec)

mysql> _
```

<u>Eclipse:</u> If the parameter **(StudentID="S112")** of Object is not match with database's record then eclipse shows an exception:



```
<terminated> RetrievingRecord [Java Application] C:\Users\Pratapgad\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.1.v202404
Jul 29, 2024 6:43:39 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledCor
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Jul 29, 2024 6:43:40 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
Jul 29, 2024 6:43:41 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl ge
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnviror
Jul 29, 2024 6:43:41 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlat
Hibernate:
    select
        student0_.StudentId as studenti1_0_0_,
        student0_.DateOfBirth as dateofbi2_0_0_,
        student0_.Email as email3_0_0_,
        student0_.FirstName as firstnam4_0_0_,
        student0_.Gender as gender5_0_0_,
        student0_.LastName as lastname6_0_0_,
        student0_.Phone as phone7_0_0_
    from
        Student student0_
    where
        student0_.StudentId=?
Exception in thread "main" org.hibernate.ObjectNotFoundException: No row with the given identifier exists: [com.demo
        at org.hibernate.boot.internal.StandardEntityNotFoundDelegate.handleEntityNotFound(StandardEntityNotFoundDel
        at org.hibernate.proxy.AbstractLazyInitializer.checkTargetState(AbstractLazyInitializer.java:298)
        at org.hibernate.proxy.AbstractLazyInitializer.initialize(AbstractLazyInitializer.java:187)
        at org.hibernate.proxy.AbstractLazyInitializer.getImplementation(AbstractLazyInitializer.java:322)
        at org.hibernate.proxy.pojo.bytebuddy.ByteBuddyInterceptor.intercept(ByteBuddyInterceptor.java:45)
        at org.hibernate.proxy.ProxyConfiguration$InterceptorDispatcher.intercept(ProxyConfiguration.java:95)
        at com.demo.college_management_system.Student$HibernateProxy$vF70qRPO.toString(Unknown Source)
        at java.base/java.lang.String.valueOf(String.java:4507)
        at java.base/java.io.PrintStream.println(PrintStream.java:1187)
        at RetrievingRecord.main(RetrievingRecord.java:18)
```

<u>Eclipse:</u> If the parameter **(StudentID="S111")** of Object is match with database's record then eclipse ran the program successfully and shows the record:



```
<terminated> RetrievingRecord [Java Application] C:\Users\Pratapgad\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.1.v202404
INFO: HHH000412: Hibernate ORM core version 5.6.15.Final
Jul 29, 2024 6:47:38 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
Jul 29, 2024 6:47:39 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)
Jul 29, 2024 6:47:39 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCrea
INFO: HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/college_management_sy
Jul 29, 2024 6:47:39 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCrea
INFO: HHH10001001: Connection properties: {password=****, user=root}
Jul 29, 2024 6:47:39 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCrea
INFO: HHH10001003: Autocommit mode: false
Jul 29, 2024 6:47:39 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledCor
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Jul 29, 2024 6:47:39 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
Jul 29, 2024 6:47:40 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl ge
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnviror
Jul 29, 2024 6:47:40 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlat
Hibernate:
    select
        student0_.StudentId as studenti1_0_0_,
        student0_.DateOfBirth as dateofbi2_0_0_,
        student0_.Email as email3_0_0_,
        student0_.FirstName as firstnam4_0_0_,
        student0_.Gender as gender5_0_0_,
        student0_.LastName as lastname6_0_0_,
        student0_.Phone as phone7_0_0_
    from
        Student student0_
    where
        student0_.StudentId=?
Student [studentId=S111, firstName=Oliver,lastName=Henry, dateOfBirth=1988-01-13, gender=M,email=oliver@gmail.com, p
```