FPT Education
FPT UNIVERSITY

Java

Java Persistence API

Objectives

- Explain what JPA is and its role in data persistence for Java applications.
- Highlight the advantages of using JPA over traditional JDBC approaches.
- Define and illustrate core JPA concepts like entities, annotations, persistence context, entity manager and JPQL.
- Provide a clear understanding of how these components work together to manage data.
- Show how JPA simplifies development by reducing boilerplate code and allo wing focus on business logic.

FPT Education
FPT UNIVERSITY

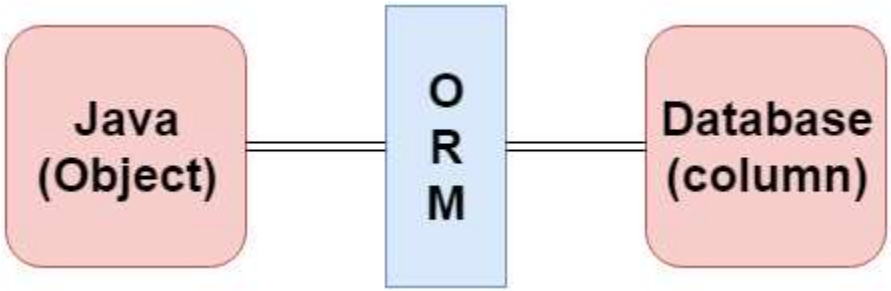
Contents

- Introduction
- Key Concepts
- Annotations
- Relationships
- Using JPA
- Demo
- Advantages and Disadvantages

Object Relational Mapping (ORM)

JPA Object Relational Mapping

- ORM makes it easier to work with databases in object-oriented applications, allowing you to focus on the business logic rather than the underlying data storage details.



Benefits of ORM

- **Increased Productivity:** You write less code because ORM handles the low-level data access tasks.
- **Improved Maintainability:** Your code is cleaner and easier to understand because it focuses on the business logic, not database details.
- **Enhanced Performance:** ORM frameworks can optimize data access and caching, potentially improving performance.
- **Database Independence:** You can switch databases without rewriting your a pplication code (to some extent)

Popular ORM Frameworks for Java

- **Hibernate:** The most widely used JPA implementation, offering a rich feature set and extensive customization options.
- **EclipseLink:** Another JPA implementation with a focus on standards complia nce and portability.
- **Spring Data JPA:** An abstraction layer on top of JPA providers, simplifying d ata access and reducing boilerplate code.
- **DataNucleus** providing easy persistence to RDBMS datastores. Comes with its own "SQL-like" JPQL query language, so you query your data in a language similar to what your datastore understands.

FPT Education
FPT UNIVERSITY

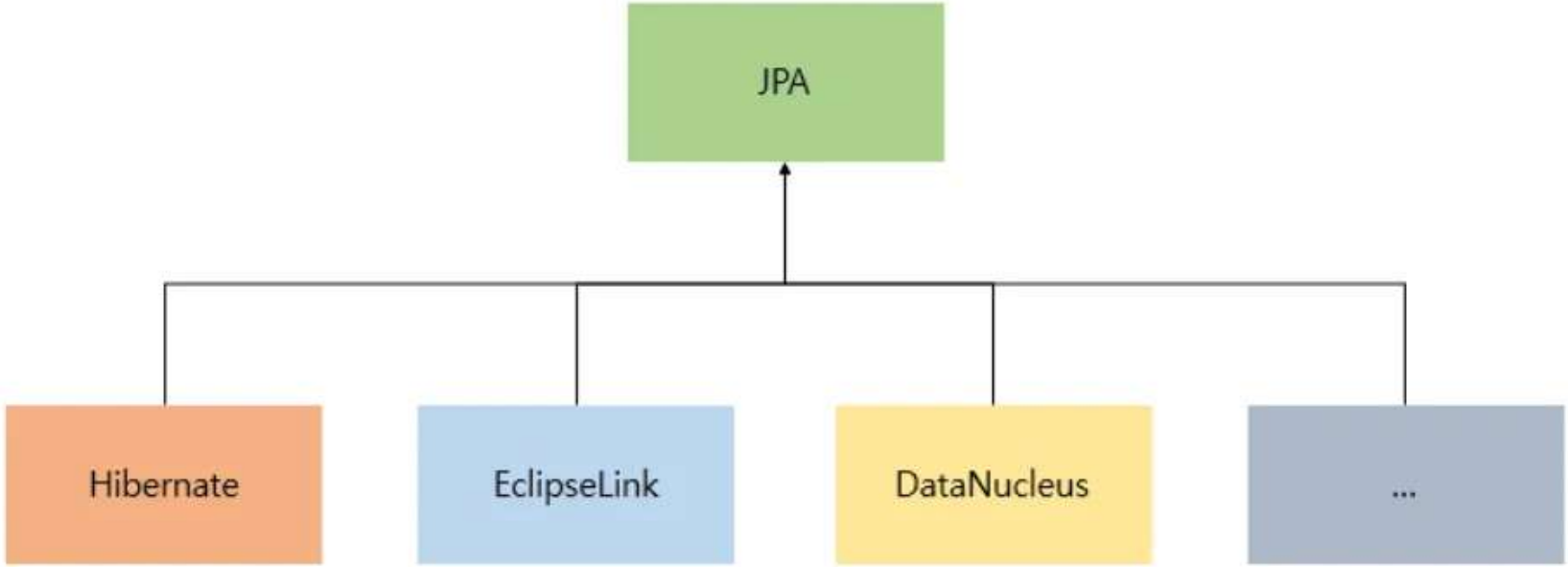
Java

Java Persistence API (JPA)

JPA Versions and Key Features

JPA Version	Release Year	Key Features
JPA 1.0	2006	Core ORM functionality: entities, mappings, relationships, inheritance, JPQL, EntityManager
JPA 2.0	2009	Enums, Criteria API, embeddable collections, derived properties, validation, metamodel
JPA 2.1	2013	Entity graphs, converters, stored procedures, Java 8 date/time
JPA 2.2	2019	Streamlined bootstrap, Java 9 modules
JPA 3.0	2022	Renamed to Jakarta Persistence, Java records, Java 17 support

JPA Implementation Options



What is JPA ?

- JPA stands for Java Persistence API. It is a Java programming interface that allows developers to manage relational data in Java applications using object oriented methodologies.
- JPA is a specification for managing relational data in Java applications. It provides a set of interfaces, annotations, and an object relational mapping (ORM) framework that simplifies the process of interacting with databases.

Key Feature of JPA

- **ORM**: JPA allows you to map Java objects (entities) to relational database tables, abstracting away the differences between the object-oriented and relational models.
- **Annotations**: You can use annotations to define entities, relationships and other persistence related metadata.
- **EntityManager**: The interface provides methods for persisting, retrieving, updating, and deleting entities.
- **JPQL** is an object oriented query language that allows you to query entities and their relationships.
- **Transaction Management**: JPA supports transaction management, ensuring data integrity and consistency.

Benefits of Using JPA

- **Simplified Data Access:** JPA makes it easier to work with relational data in Ja va applications by providing a higher-level abstraction.
- **Code Portability:** JPA is a standard specification, so code written using JPA can be portable across different JPA providers.
- **Reduced Boilerplate Code:** JPA annotations and the EntityManager interface reduce the amount of code needed for data access.
- **Improved Data Integrity:** JPA's transaction management and validation features help ensure data integrity.
- **Enhanced Performance:** Caching and other optimization techniques can improve the performance of JPA applications.

Entity Annotations in JPA

- **@Entity**: This annotation marks a Java class as an entity, meaning it represents a table in the database.
- **@Table:** This annotation allows you to specify the name of the database table that an entity maps to.
- **@Transient:** This annotation marks a property that should not be persisted to the database.
- **@NamedQueries and @NamedQuery:** These annotations allow you to define named queries that can be used to retrieve entities.

Mapping Annotations in JPA

- **@Id**: This annotation marks a Java class as an entity, meaning it represents a table in the database.
- **@Column:** This annotation defines the mapping between a property of an entity and a column in the database table. You can use it to specify the column name, data type, and other attributes.
- **@Basic:** This annotation specifies that a property is a basic type and should be persisted.
- **@Enumerated:** This annotation is used to map an Enum type to a database column.

JPA Entity Manager

- The entity manager implements the API and encapsulates all of them within a single interface.
  - **Manages entity lifecycle:** Persist, find, merge, remove
  - **Controls persistence context:** Cache, flush, detach
  - **Executes queries:** JPQL, Criteria API
  - **Handles transactions:** Begin, commit, rollback
  - **Provides access to entity metadata**

FPT Education
FPT UNIVERSITY

Key Operations of the Entity Manager

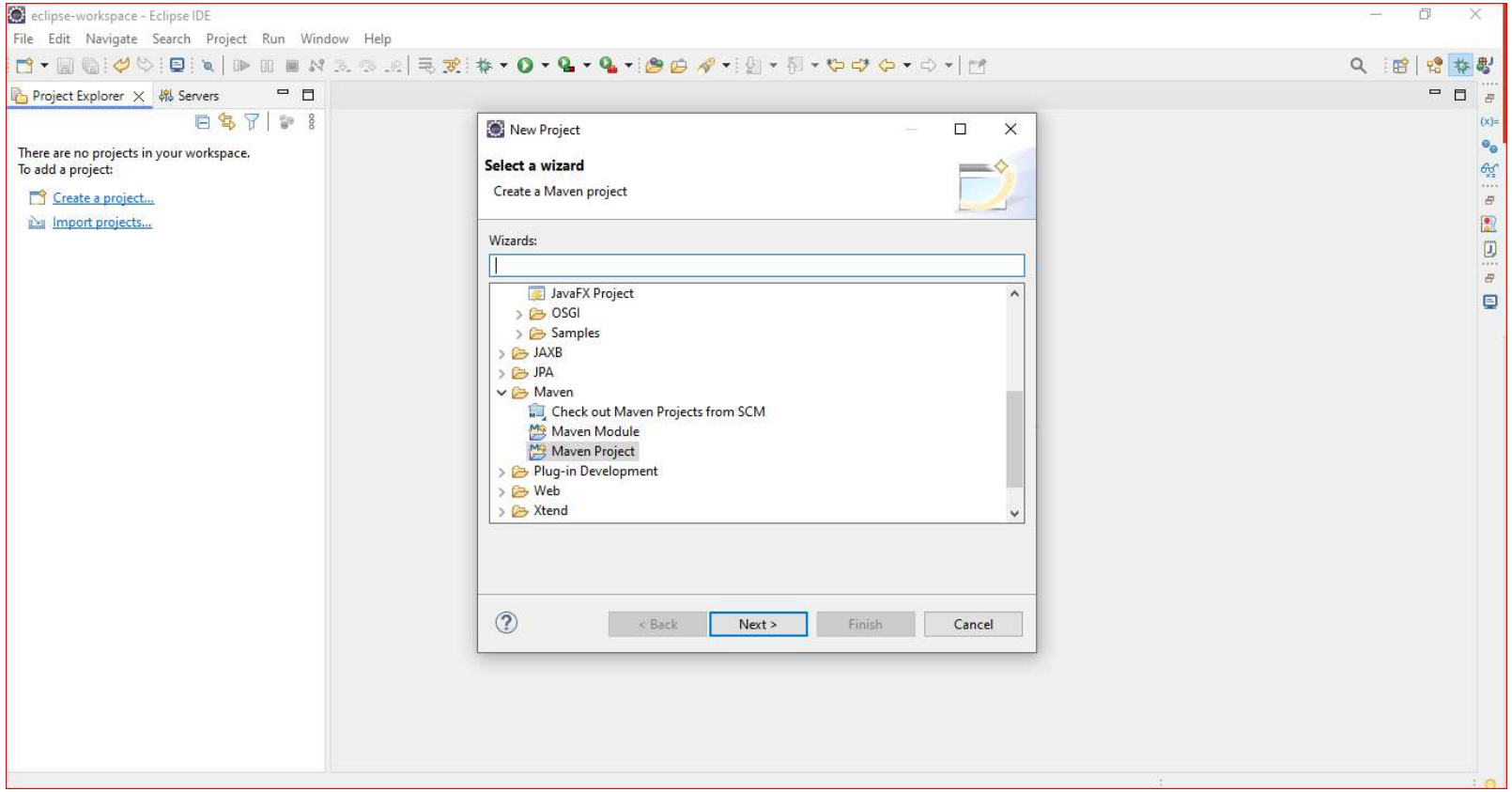
- **Persisting Entities:** *em.persist(entity)* makes a transient entity instance persistent
- **Finding Entities:** *em.find(entityClass,primaryKey)* retrieves an entity by its primary key.
- **Merging Entities:** *em.merge(entity)* merges a detached entity instance into the current persistence context.
- **Removing Entities:** *em.remove(entity)* removes a persistent entity instance.
- **Querying Data:** *em.createQuery(jpqlString)* creates a JPQL query to retrieve entities based on specified criteria.

FPT Education
FPT UNIVERSITY

Java

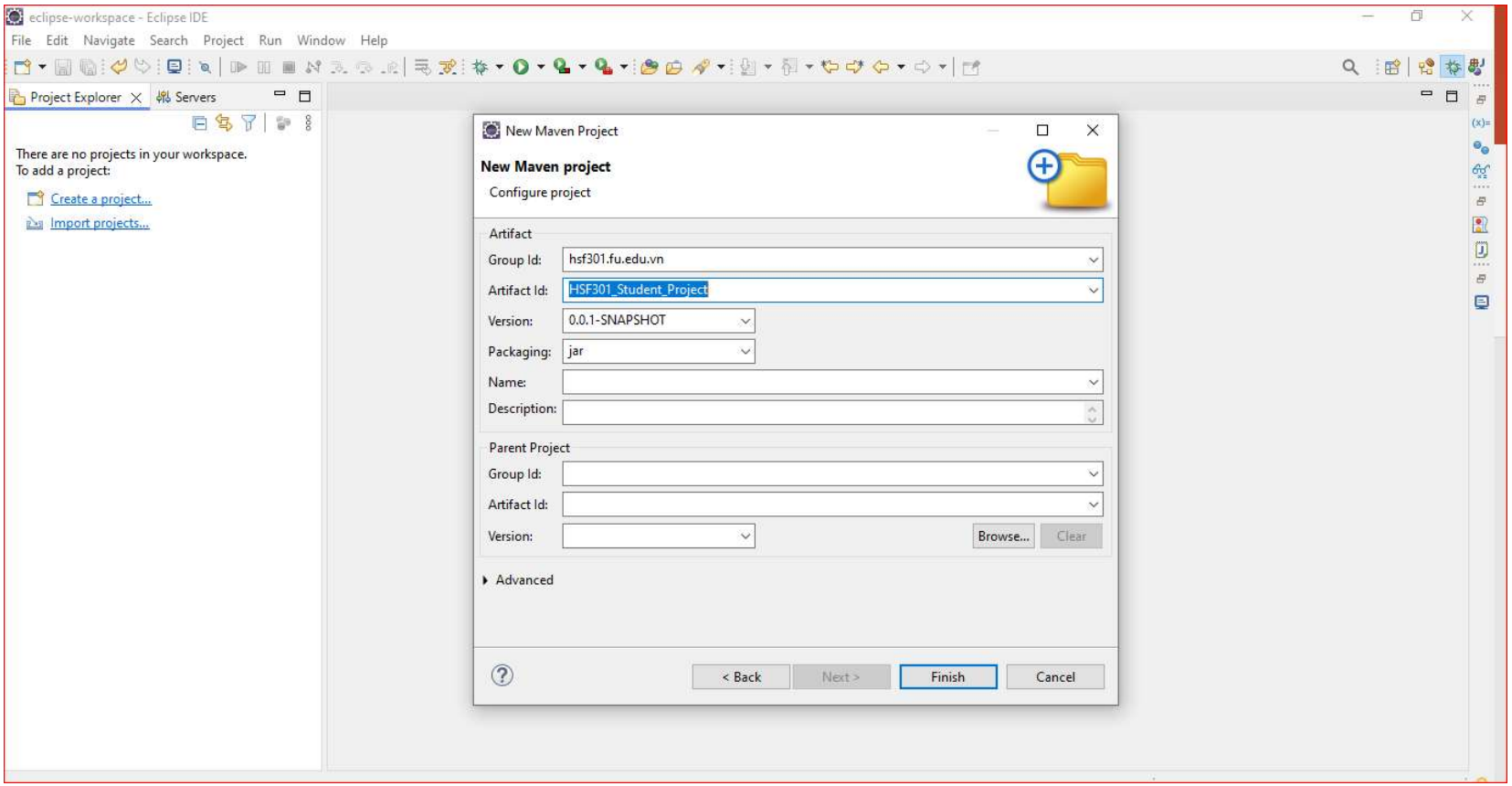
Demo Simple CRUD with JPA

1. Open Eclipse, File | New | Maven Project

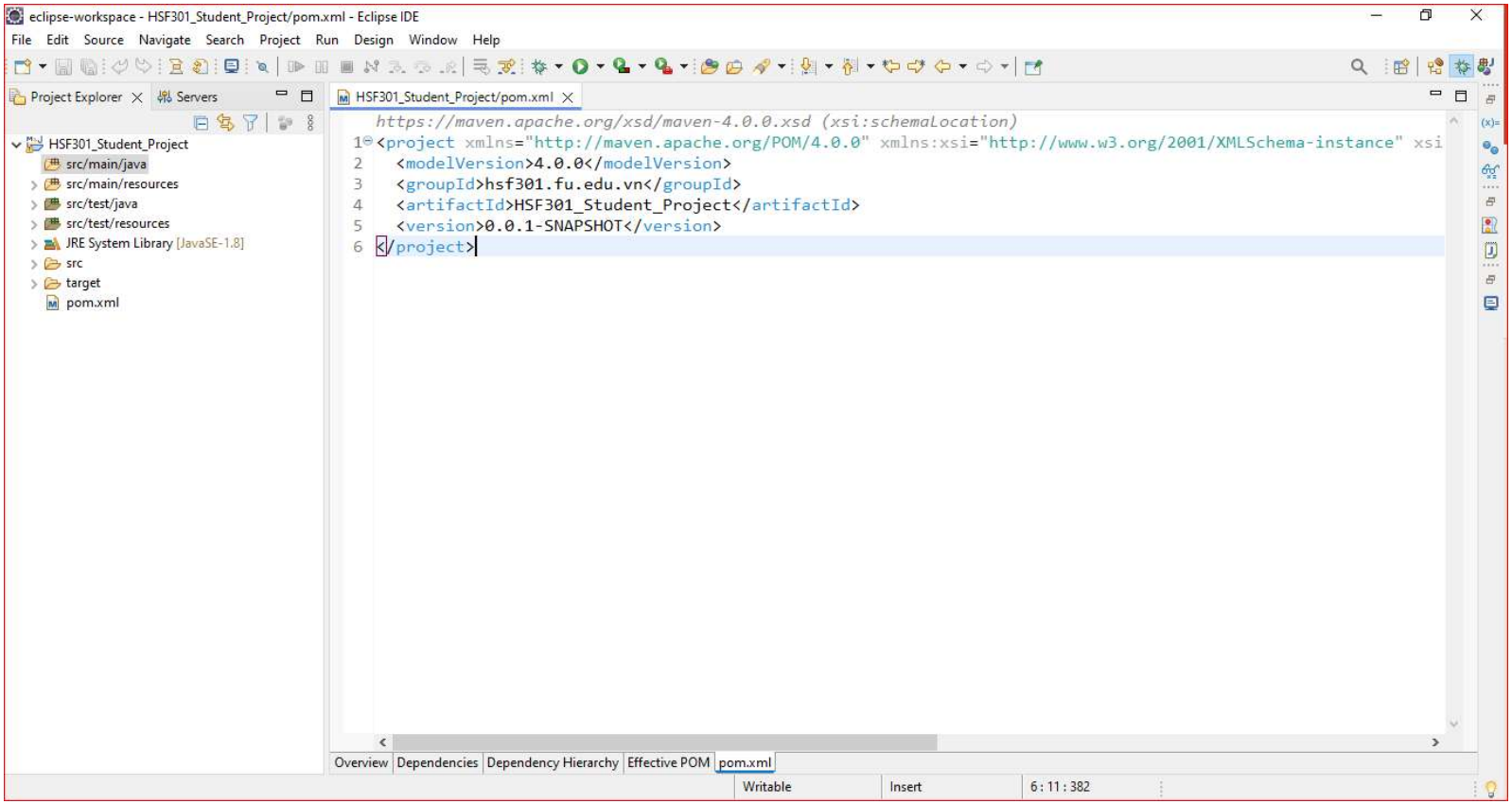


FPT Education
FPT UNIVERSITY

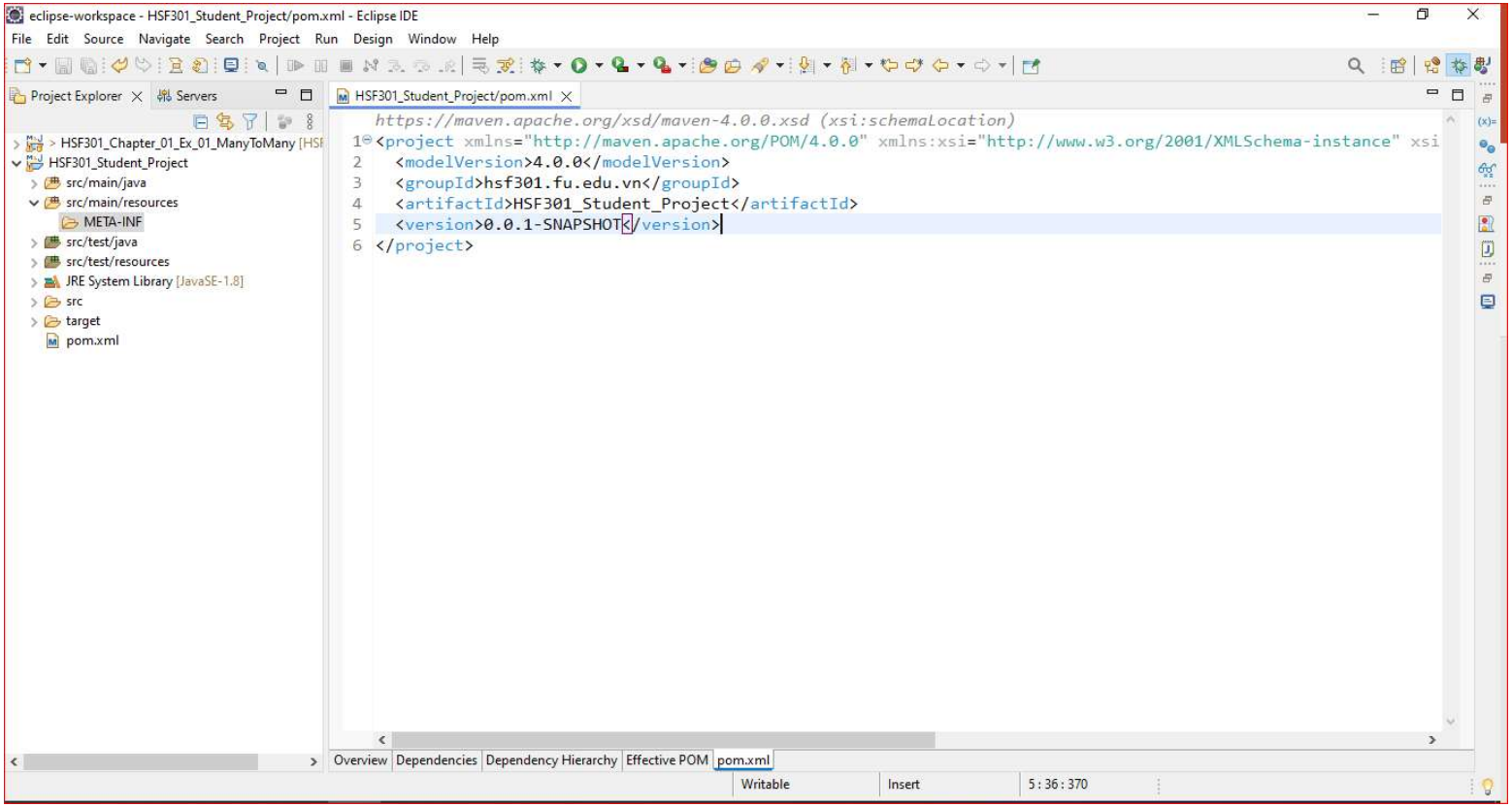
2. Check Create a simple project -> Browse Project -> Next



3. The New Project



4. Create folder META-INF in src/main/resources



5. Edit the pom.xml

eclipse-workspace - HSF301_Student_Project/pom.xml - Eclipse IDE

File Edit Source Navigate Search Project Run Design Window Help

Project Explorer | Servers

- HSF301_Chapter_01_Ex_01_ManyToMany [HSF301 main]
- HSF301_Student_Project
  - src/main/java
  - src/main/resources
    - META-INF
  - src/test/java
  - src/test/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

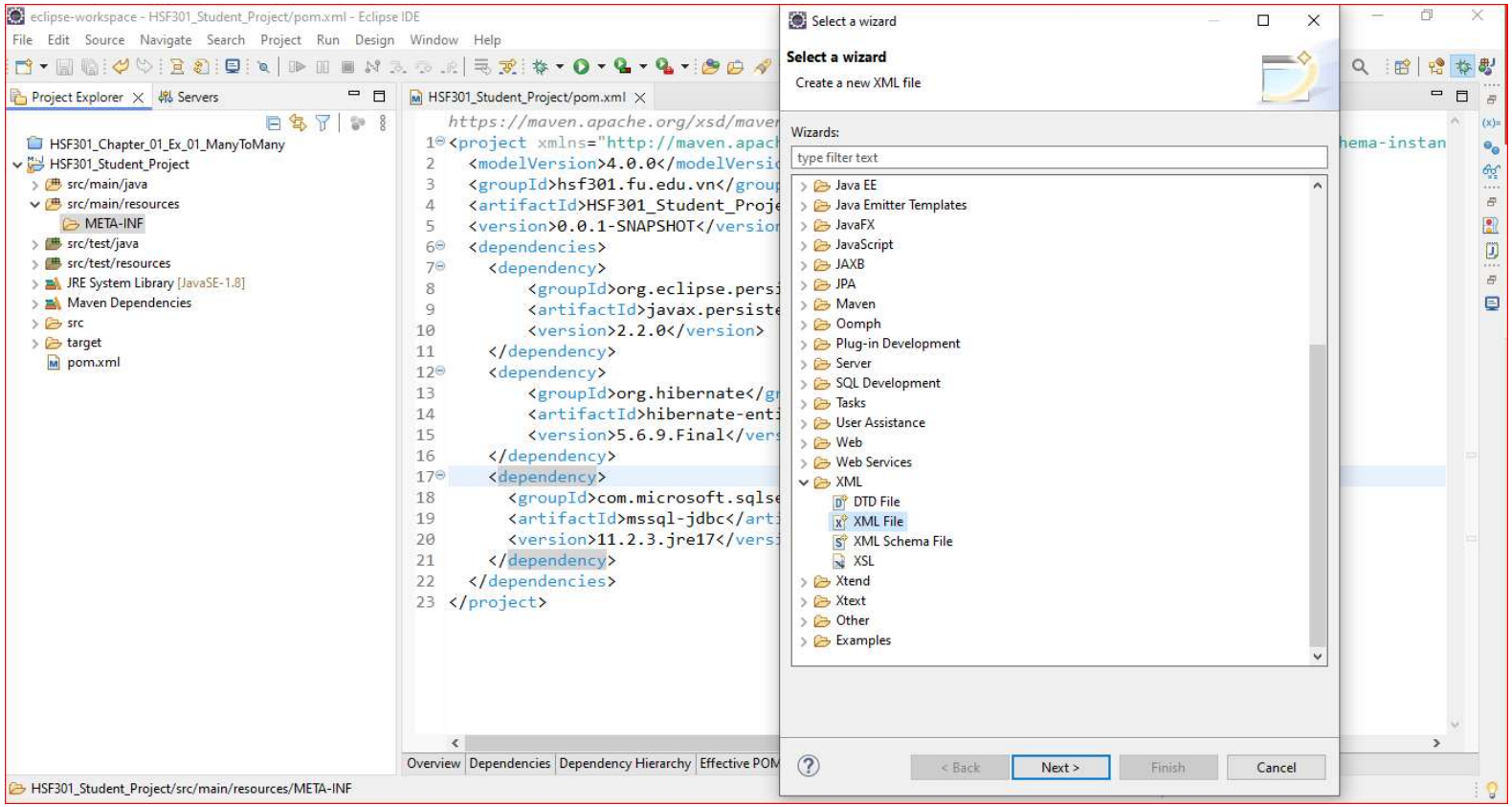
HSF301_Student_Project/pom.xml

```
  https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instan
2   <modelVersion>4.0.0</modelVersion>
3   <groupId>hsf301.fu.edu.vn</groupId>
4   <artifactId>HSF301_Student_Project</artifactId>
5   <version>0.0.1-SNAPSHOT</version>
6   <dependencies>
7     <dependency>
8         <groupId>org.eclipse.persistence</groupId>
9         <artifactId>javax.persistence</artifactId>
10        <version>2.2.0</version>
11    </dependency>
12    <dependency>
13        <groupId>org.hibernate</groupId>
14        <artifactId>hibernate-entitymanager</artifactId>
15        <version>5.6.9.Final</version>
16    </dependency>
17    <dependency>
18      <groupId>com.microsoft.sqlserver</groupId>
19      <artifactId>mssql-jdbc</artifactId>
20      <version>11.2.3.jre17</version>
21    </dependency>
22  </dependencies>
23 </project>
```

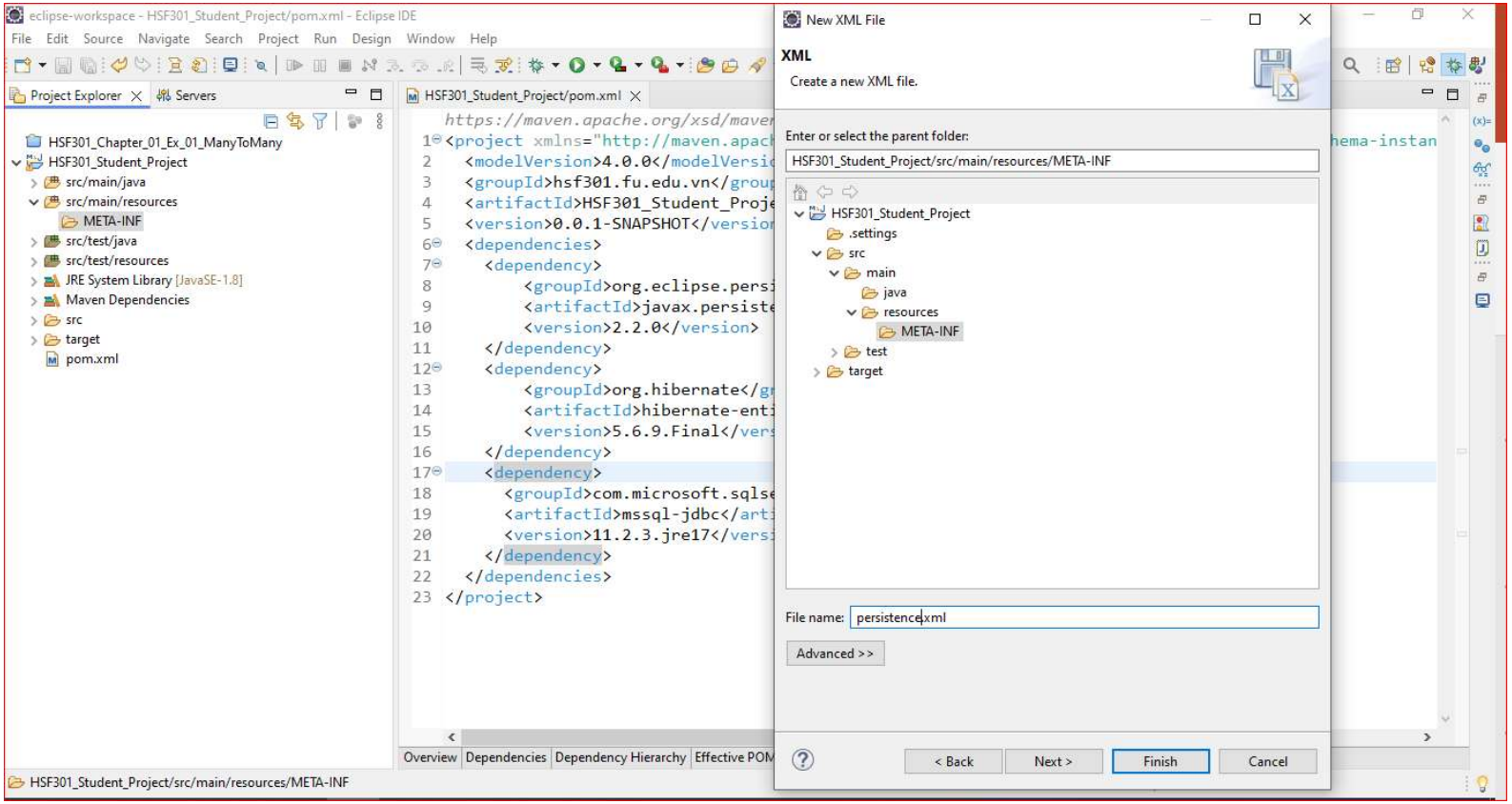
Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

Writable | Insert | 23 : 11 : 898

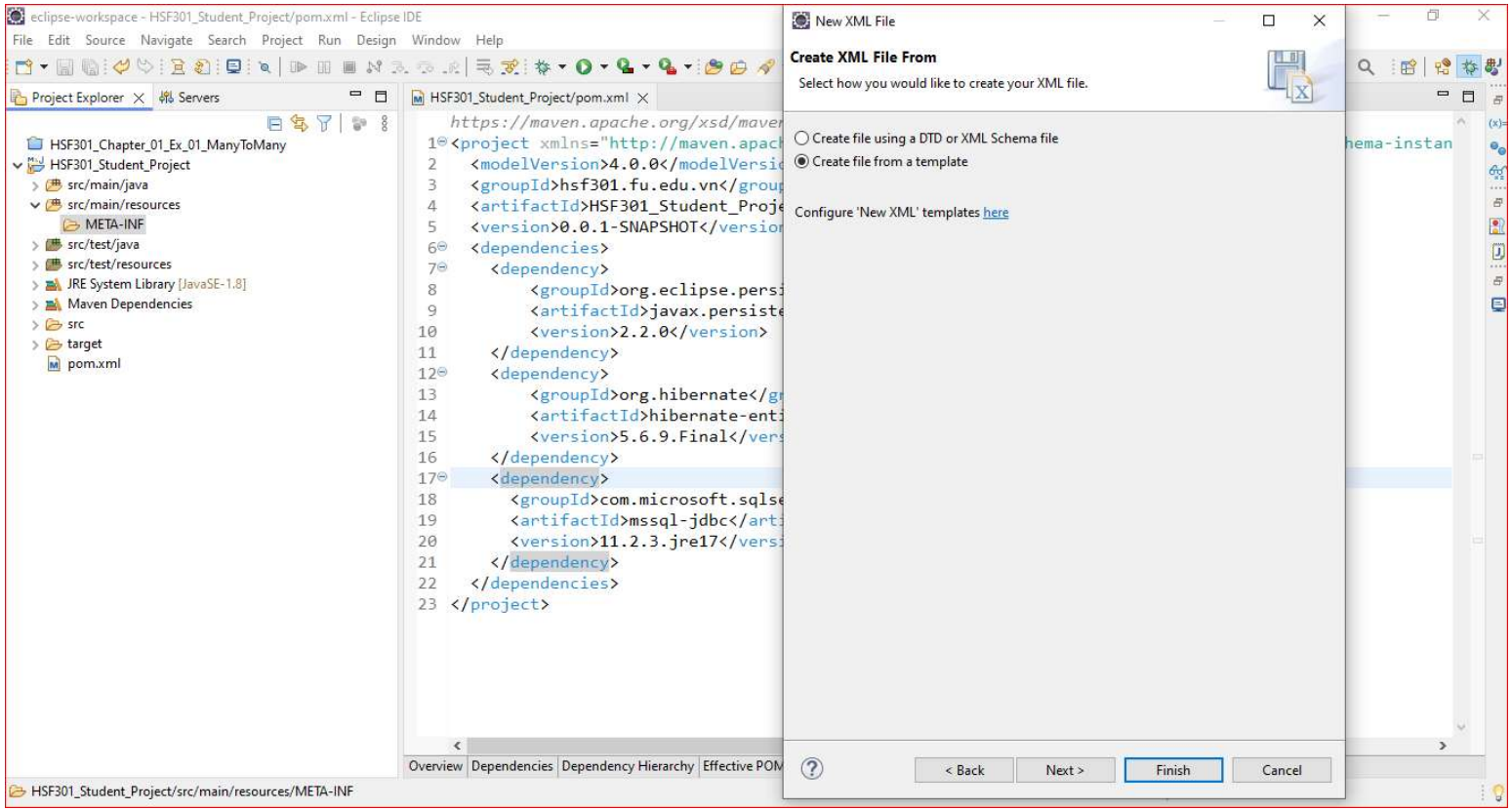
6. Create persistence.xml in META-INF folder



7. Create persistence.xml in META-INF folder



8. Create persistence.xml in META-INF folder



9. Edit the persistence.xml in META-INF folder

eclipse-workspace - HSF301_Student_Project/src/main/resources/META-INF/persistence.xml - Eclipse IDE

File Edit Source Source Navigate Search Project Run Window Help

Project Expl... | Servers

- HSF301_Chapter_01_Ex_01_ManyToMar
- HSF301_Student_Project
  - src/main/java
  - src/main/resources
    - META-INF
      - persistence.xml
  - src/test/java
  - src/test/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

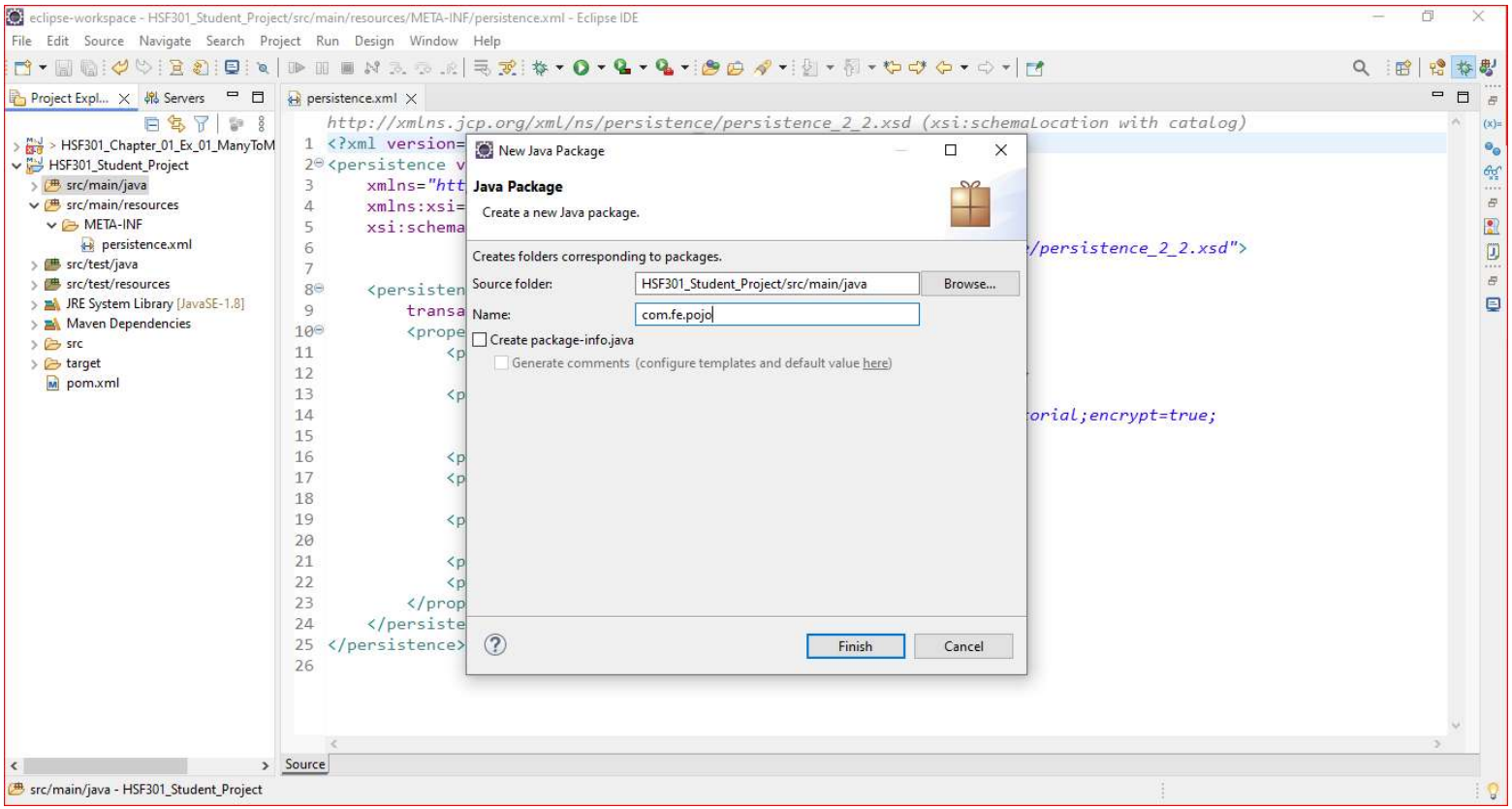
persistence.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.2"
    xmlns="http://xmlns.jcp.org/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
                        http://xmlns.jcp.org/xml/ns/persistence/persistence_2_2.xsd">

    <persistence-unit name="JPAs"
        transaction-type="RESOURCE_LOCAL">
        <properties>
            <property name="javax.persistence.jdbc.driver"
                value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
            <property name="javax.persistence.jdbc.url"
                value="jdbc:sqlserver://localhost:1433;DatabaseName=tutorial;encrypt=true;
                trustServerCertificate=true;" />
            <property name="javax.persistence.jdbc.user" value="sa" />
            <property name="javax.persistence.jdbc.password"
                value="Aesx5099@@" />
            <property name="hibernate.dialect"
                value="org.hibernate.dialect.SQLServerDialect" />
            <property name="hibernate.show_sql" value="true" />
            <property name="hibernate.hbm2ddl.auto" value="update" />
        </properties>
    </persistence-unit>
</persistence>
```

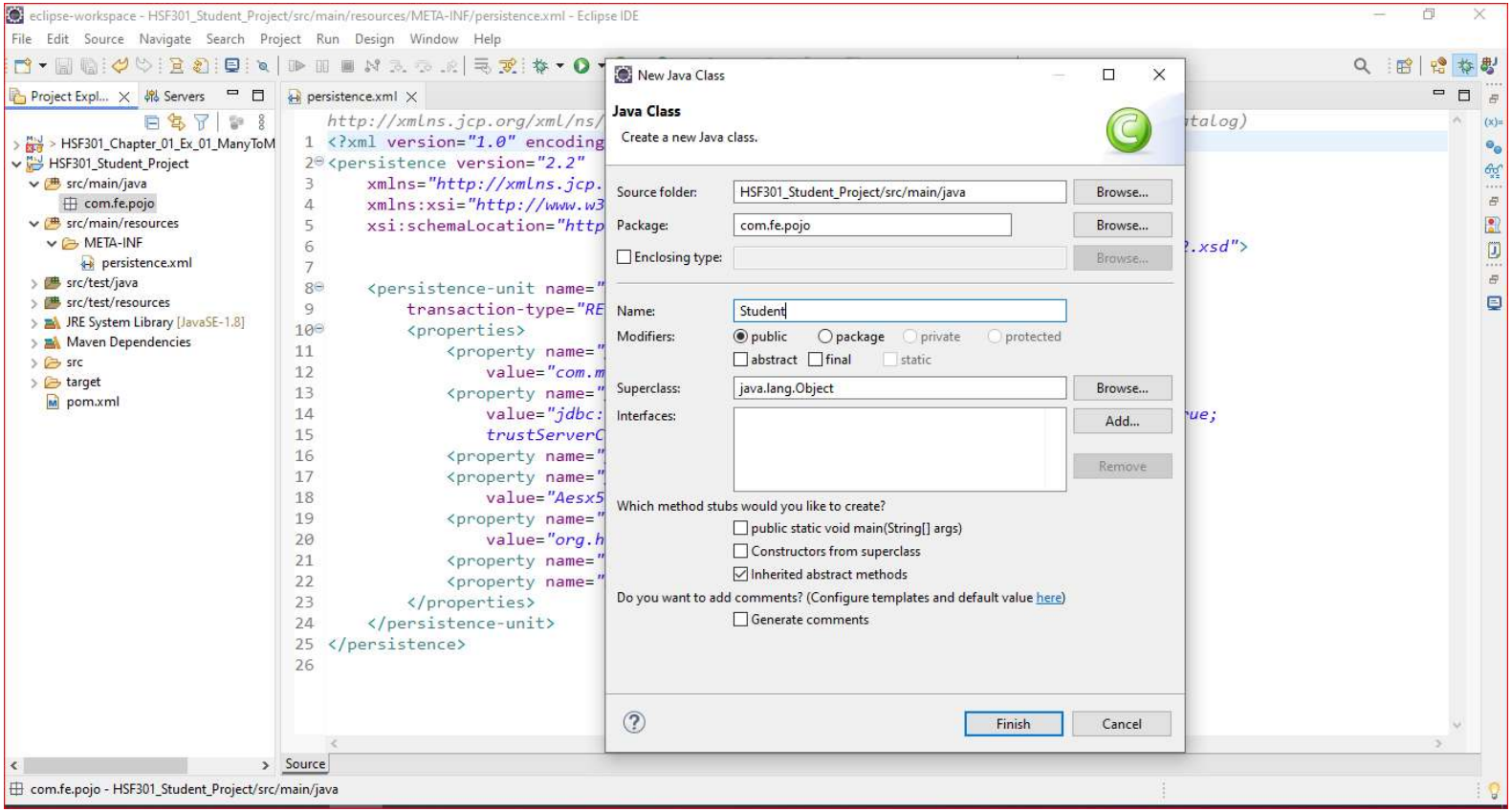
Design | Source

persistence/persistence-unit | Writable | Smart Insert | 24 : 22 : 1120

10. Add com.fe.pojo Package in src/main/java



11. Add Student.java in com.fe.pojo package



12. Edit the Student.java

eclipse-workspace - HSF301_Student_Project/src/main/java/com/fe/pojo/Student.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Expl... | Servers

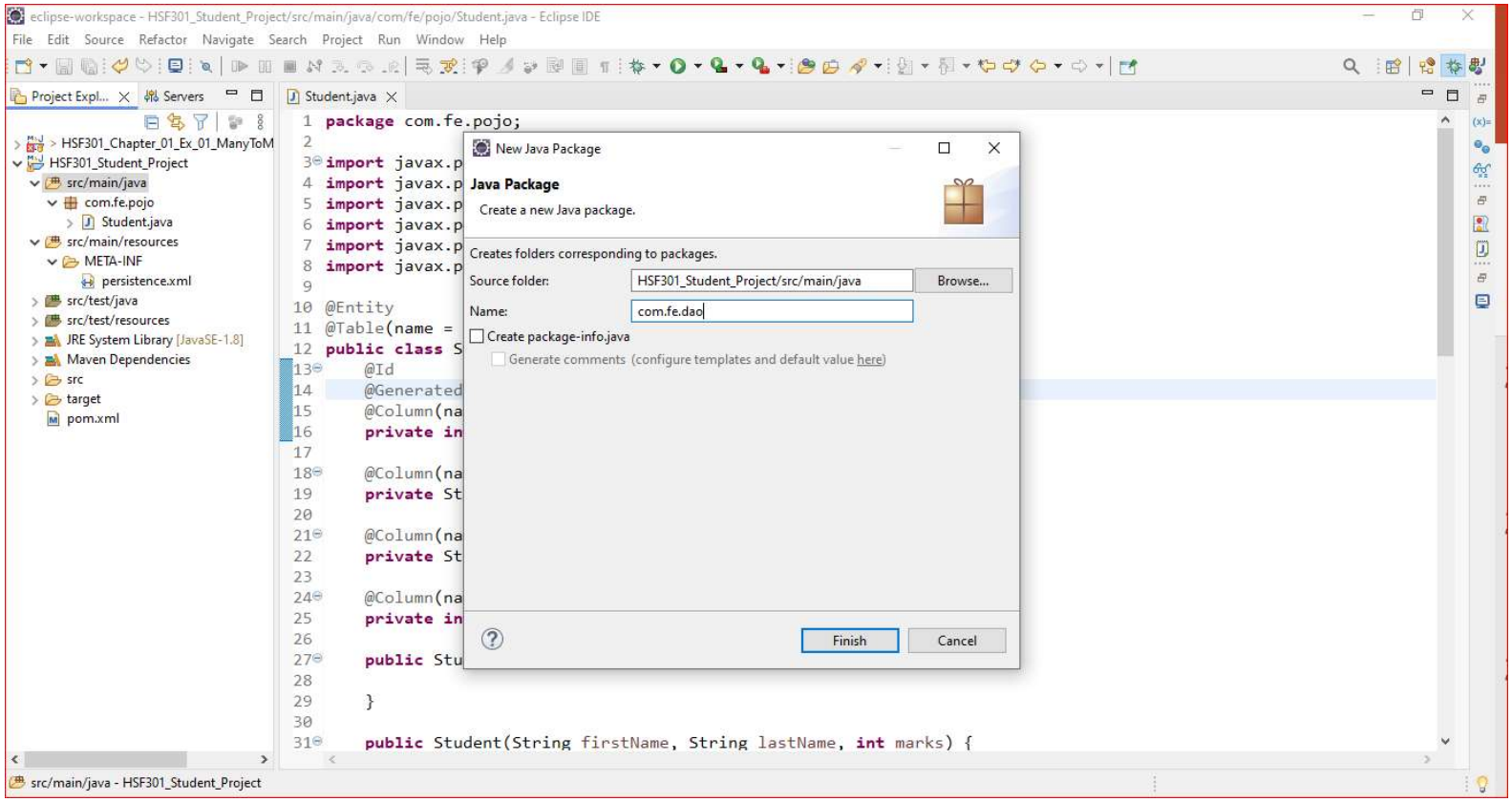
- HSF301_Chapter_01_Ex_01_ManyToM
- HSF301_Student_Project
  - src/main/java
    - com.fe.pojo
      - Student.java
  - src/main/resources
    - META-INF
      - persistence.xml
  - src/test/java
  - src/test/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Student.java

```java
package com.fe.pojo;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "STUDENTS")
public class Student {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private int id;

    @Column(name = "fistName", nullable = false, unique = false)
    private String firstName;

    @Column(name = "lastName")
    private String lastName;

    @Column(name = "marks")
    private int marks;

    public Student() {

    }

    public Student(String firstName, String lastName, int marks) {
```

0 items selected

13. Add com.fe.dao Package in src/main/java



14. Create StudentDAO in com.fe.dao

eclipse-workspace - HSF301_Student_Project/src/main/java/com/fe/pojo/Student.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Expl... | Servers

- HSF301_Chapter_01_Ex_01_ManyToM
- HSF301_Student_Project
  - src/main/java
    - com.fe.dao
    - com.fe.pojo
      - Student.java
  - src/main/resources
    - META-INF
      - persistence.xml
  - src/test/java
  - src/test/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Student.java

```
1  package com.fe.pojo;
2
3  import javax.persistence.Col
4  import javax.persistence.Ent
5  import javax.persistence.Gen
6  import javax.persistence.Gen
7  import javax.persistence.Id;
8  import javax.persistence.Tab
9
10 @Entity
11 @Table(name = "STUDENTS")
12 public class Student {
13     @Id
14     @GeneratedValue(strategy
15     @Column(name = "id")
16     private int id;
17
18     @Column(name = "fistName
19     private String firstName
20
21     @Column(name = "lastName
22     private String lastName;
23
24     @Column(name = "marks")
25     private int marks;
26
27     public Student() {
28
29     }
30
31     public Student(String fi
```

New Java Class

Java Class

Create a new Java class.

Source folder: HSF301_Student_Project/src/main/java [Browse...]

Package: com.fe.dao [Browse...]

☐ Enclosing type: [Browse...]

Name: StudentDAO

Modifiers: ◉ public ○ package ○ private ○ protected

☐ abstract ☐ final ☐ static

Superclass: java.lang.Object [Browse...]

Interfaces: [Add...] [Remove]

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

☐ Generate comments

[Finish] [Cancel]

com.fe.dao - HSF301_Student_Project/src/main/java

15. Edit the StudentDAO in com.fe.dao

eclipse-workspace - HSF301_Student_Project/src/main/java/com/fe/dao/StudentDAO.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Expl... | Servers

- HSF301_Chapter_01_Ex_01_ManyToM
- HSF301_Student_Project
  - src/main/java
    - com.fe.dao
      - StudentDAO.java
    - com.fe.pojo
      - Student.java
  - src/main/resources
    - META-INF
      - persistence.xml
  - src/test/java
  - src/test/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Student.java | StudentDAO.java

```java
1  package com.fe.dao;
2
3  import java.util.List;
4
5  import javax.persistence.EntityManager;
6  import javax.persistence.EntityManagerFactory;
7  import javax.persistence.Persistence;
8
9  import com.fe.pojo.Student;
10
11 public class StudentDAO {
12
13     private static EntityManager em;
14     private static EntityManagerFactory emf;
15
16     public StudentDAO(String persistanceName) {
17         emf = Persistence.createEntityManagerFactory(persistanceName);
18     }
19
20     public void save(Student student) {
35
36     public List<Student> getStudents() {
51
52     public void delete(Long studentID) {
65
66     public Student findById(Long studentID) {
80
81     public void update(Student student) {
98 }
99
```

Writable | Smart Insert | 20 : 1 : 437

FPT Education
FPT UNIVERSITY

16. Save Student Method

eclipse-workspace - HSF301_Student_Project/src/main/java/com/fe/dao/StudentDAO.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Expl... | Servers | Student.java | StudentDAO.java

- HSF301_Chapter_01_Ex_01_ManyToM
- HSF301_Student_Project
  - src/main/java
    - com.fe.dao
      - StudentDAO.java
    - com.fe.pojo
      - Student.java
  - src/main/resources
    - META-INF
      - persistence.xml
  - src/test/java
  - src/test/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

```java
3  import java.util.List;
10
11 public class StudentDAO {
12
13     private static EntityManager em;
14     private static EntityManagerFactory emf;
15
16     public StudentDAO(String persistanceName) {
17         emf = Persistence.createEntityManagerFactory(persistanceName);
18     }
19
20     public void save(Student student) {
21         try {
22             em = emf.createEntityManager();
23             em.getTransaction().begin();
24             em.persist(student);
25             em.getTransaction().commit();
26         } catch (Exception ex) {
27             em.getTransaction().rollback();
28             System.out.println("Error " + ex.getMessage());
29         } finally {
30             em.close();
31
32         }
33
34     }
35
36     public List<Student> getStudents() {
51
52     public void delete(int studentID) {
65
```

Writable | Smart Insert | 24 : 23 : 566

FPT Education
FPT UNIVERSITY

17. Get All Students Method

eclipse-workspace - HSF301_Student_Project/src/main/java/com/fe/dao/StudentDAO.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Expl... | Servers

- HSF301_Chapter_01_Ex_01_ManyToM
- HSF301_Student_Project
  - src/main/java
    - com.fe.dao
      - StudentDAO.java
    - com.fe.pojo
      - Student.java
  - src/main/resources
    - META-INF
      - persistence.xml
  - src/test/java
  - src/test/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Student.java | StudentDAO.java

```java
package com.fe.dao;

import java.util.List;

public class StudentDAO {

    private static EntityManager em;
    private static EntityManagerFactory emf;

    public StudentDAO(String persistanceName) {
        emf = Persistence.createEntityManagerFactory(persistanceName);
    }

    public void save(Student student) {

    public List<Student> getStudents() {
        List<Student> students = null;
        try {
            em = emf.createEntityManager();
            em.getTransaction().begin();
            students = em.createQuery("from Student").getResultList();
        } catch (Exception ex) {

            System.out.println("Error " + ex.getMessage());
        } finally {
            em.close();
        }
        return students;

    }

```

Writable | Smart Insert | 24 : 23 : 566

FPT Education
FPT UNIVERSITY

18. Find a Student Method

eclipse-workspace - HSF301_Student_Project/src/main/java/com/fe/dao/StudentDAO.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Expl... | Servers | Student.java | StudentDAO.java

- HSF301_Chapter_01_Ex_01_ManyToM
- HSF301_Student_Project
  - src/main/java
    - com.fe.dao
      - StudentDAO.java
    - com.fe.pojo
      - Student.java
  - src/main/resources
    - META-INF
      - persistence.xml
  - src/test/java
  - src/test/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

```java
13      private static EntityManager em;
14      private static EntityManagerFactory emf;
15
16      public StudentDAO(String persistanceName) {
17          emf = Persistence.createEntityManagerFactory(persistanceName);
18      }
19
20      public void save(Student student) {
35
36      public Student findById(int studentID) {
37          Student student = null;
38          try {
39              em = emf.createEntityManager();
40              em.getTransaction().begin();
41              student = em.find(Student.class, studentID);
42          } catch (Exception ex) {
43              System.out.println("Error " + ex.getMessage());
44          } finally {
45              em.close();
46          }
47          return student;
48
49      }
50
51      public List<Student> getStudents() {
66
67      public void delete(int studentID) {
80
81      public void update(Student student) {
98  }
99
```

Writable | Smart Insert | 50 : 5 : 1113

19. Delete Student Method

eclipse-workspace - HSF301_Student_Project/src/main/java/com/fe/dao/StudentDAO.java - Eclipse IDE

```java
    private static EntityManager em;
    private static EntityManagerFactory emf;

    public StudentDAO(String persistanceName) {
        emf = Persistence.createEntityManagerFactory(persistanceName);
    }

    public void save(Student student) {

    public Student findById(int studentID) {

    public List<Student> getStudents() {

    public void delete(int studentID) {
        try {
            em = emf.createEntityManager();
            em.getTransaction().begin();
            Student s = em.find(Student.class, studentID);
            em.remove(s);
            em.getTransaction().commit();
        } catch (Exception ex) {
            System.out.println("Error " + ex.getMessage());
        } finally {
            em.close();
        }
    }

    public void update(Student student) {
}
```

20. Update Student Method

eclipse-workspace - HSF301_Student_Project/src/main/java/com/fe/dao/StudentDAO.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Expl... | Servers | Student.java | StudentDAO.java

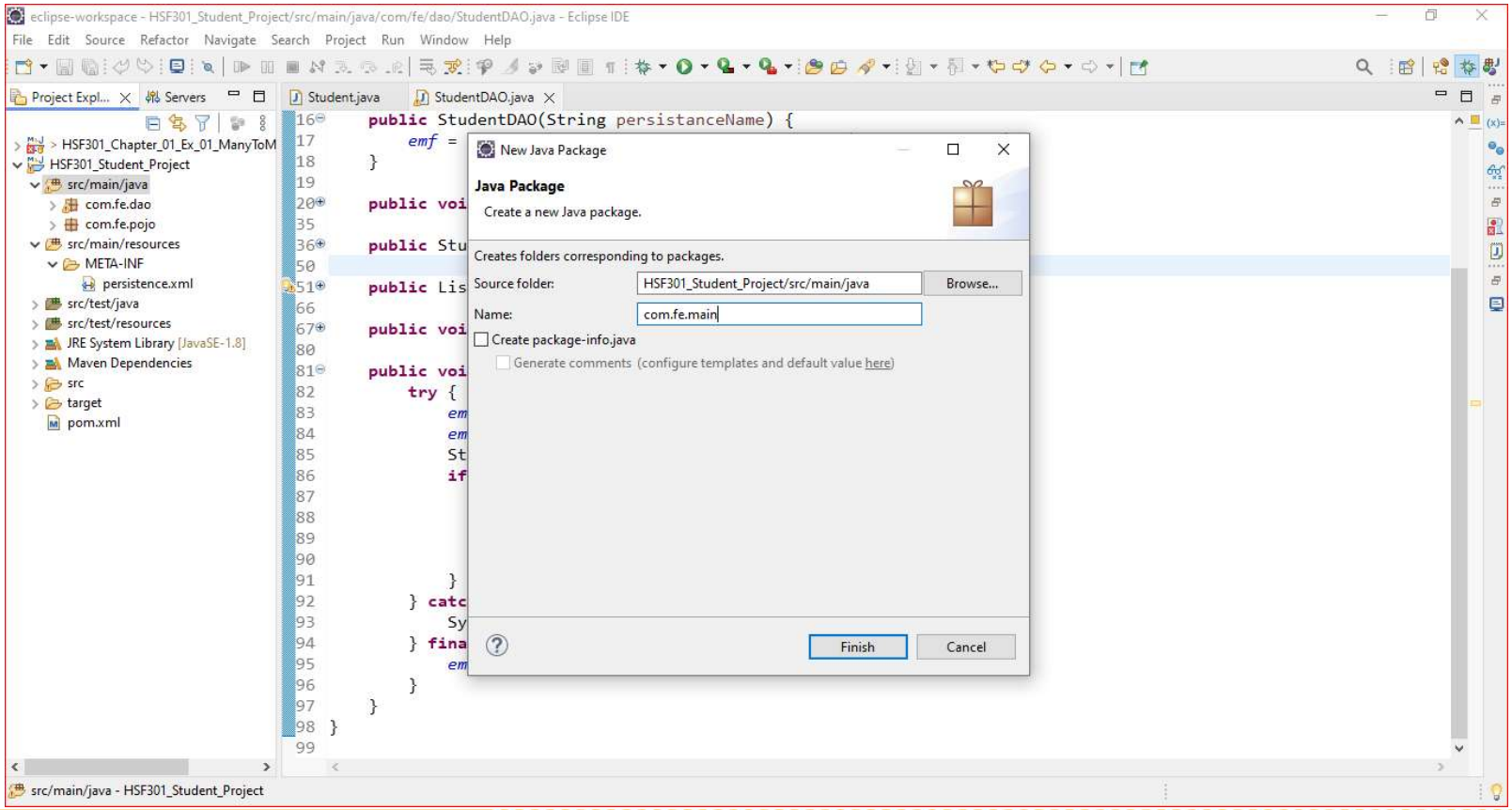
- HSF301_Chapter_01_Ex_01_ManyToM
- HSF301_Student_Project
  - src/main/java
    - com.fe.dao
      - StudentDAO.java
    - com.fe.pojo
      - Student.java
  - src/main/resources
    - META-INF
      - persistence.xml
  - src/test/java
  - src/test/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

```java
16      public StudentDAO(String persistanceName) {
17          emf = Persistence.createEntityManagerFactory(persistanceName);
18      }
19
20      public void save(Student student) {
35
36      public Student findById(int studentID) {
50
51      public List<Student> getStudents() {
66
67      public void delete(int studentID) {
80
81      public void update(Student student) {
82          try {
83              em = emf.createEntityManager();
84              em.getTransaction().begin();
85              Student s = em.find(Student.class, student.getId());
86              if (s != null) {
87                  s.setFirstName(student.getFirstName());
88                  s.setLastName(student.getLastName());
89                  s.setMarks(student.getMarks());
90                  em.getTransaction().commit();
91              }
92          } catch (Exception ex) {
93              System.out.println("Error " + ex.getMessage());
94          } finally {
95              em.close();
96          }
97      }
98  }
99
```

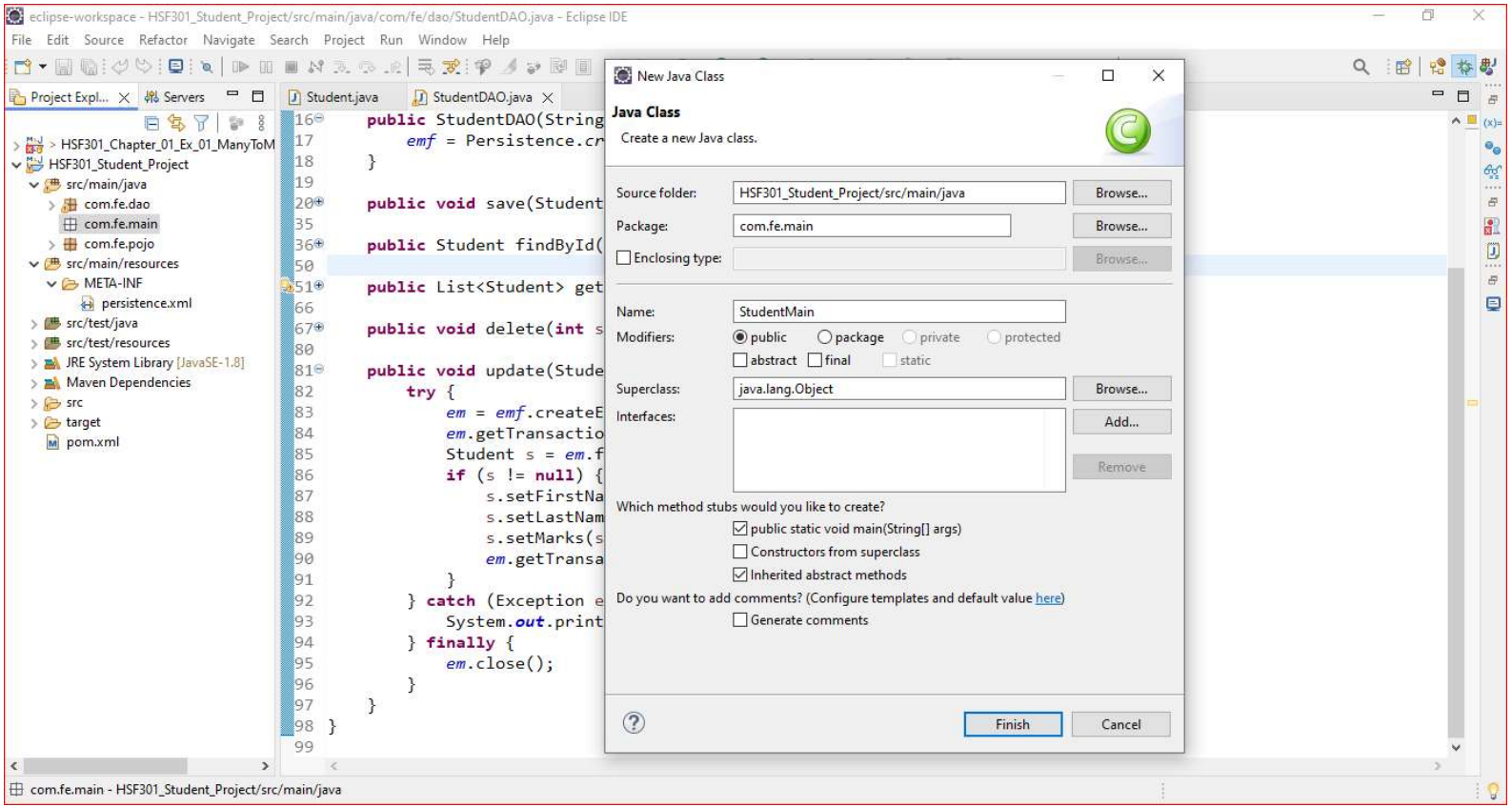
Writable | Smart Insert | 50 : 5 : 1113

FPT Education
FPT UNIVERSITY

21. Add com.fe.main Package in src/main/java



22. Create StudentMain class in com.fe.main Package



23. Edit the StudentMain in com.fe.main

eclipse-workspace - HSF301_Student_Project/src/main/java/com/fe/main/StudentMain.java - Eclipse IDE

```java
package com.fe.main;

public class StudentMain {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("+++++++++MENU+++++++++");
        System.out.println("+ 1. Get Students    +");
        System.out.println("+ 2. Add Student     +");
        System.out.println("+ 3. Delete Student  +");
        System.out.println("+ 4. Update Student  +");
        System.out.println("+ 5. Delete Student  +");
        System.out.println("+ 6. Get a Student   +");
        System.out.println("+ 0. QUIT            +");
        System.out.println("+++++++++END++++++++++");
    }

}
```

```
<terminated> StudentMain [Java Application] D:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
+++++++++MENU+++++++++
+ 1. Get Students    +
+ 2. Add Student     +
+ 3. Delete Student  +
+ 4. Update Student  +
+ 5. Delete Student  +
+ 6. Get a Student   +
+ 0. QUIT            +
+++++++++END++++++++++
```

FPT Education
FPT UNIVERSITY

24. Edit the StudentMain in com.fe.main

eclipse-workspace - HSF301_Student_Project/src/main/java/com/fe/main/StudentMain.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Expl... | Servers

- HSF301_Chapter_01_Ex_01_ManyT
  - src/main/java
    - hsf301.fe.dao
      - StudentDAO.java
    - hsf301.fe.gui
      - ManyToManyDemo.java
    - hsf301.fe.pojo
      - Book.java
      - Student.java
    - hsf301.fe.repository
    - hsf301.fe.service
  - src/main/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml
- HSF301_Student_Project
  - src/main/java
    - com.fe.dao
    - com.fe.main
      - StudentMain.java
    - com.fe.pojo
      - Student.java
  - src/main/resources
    - META-INF
      - persistence.xml
  - src/test/java
  - src/test/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies

Student.java | StudentDAO.java | StudentMain.java

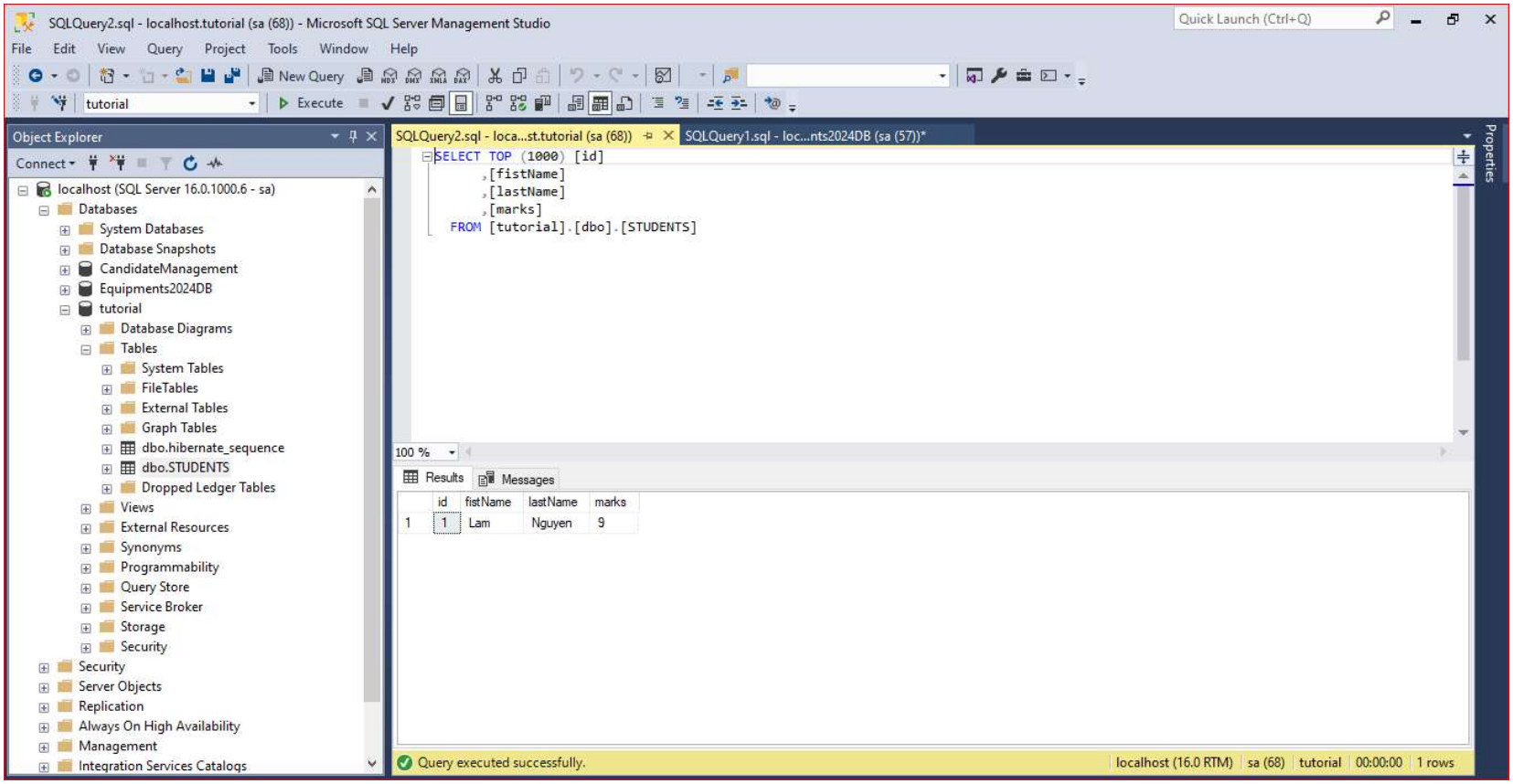
```java
        System.out.println("+ 0. QUIT            +");
        System.out.println("+++++++++END++++++++++");

        int inputKey = -1;
        while (inputKey != 0) {
            Scanner console = new Scanner(System.in);
            System.out.println("Please enter a number !");
            inputKey = console.nextInt();
            StudentDAO studentDAO = new StudentDAO("JPAs");
            Student student = new Student("Lam", "Nguyen", 9);
            switch (inputKey) {
            case 0:
                break;
            case 1:
                studentDAO.save(student);
                break;
            case 2:
                studentDAO.delete(1);
                break;
            case 3:
                student = new Student(1, "Sang", "Nguyen", 9);
                studentDAO.update(student);
            case 4:
                studentDAO.findById(1);
                break;
            default:
                System.out.println("Please choice menu !");
            }
        }

    }
```

Console

StudentMain [Java Application] [pid: 17148]

```
+++++++++MENU++++++++
+ 1. Add Student     +
+ 2. Delete Student  +
+ 3. Update Student  +
+ 4. Get a Student   +
+ 0. QUIT            +
+++++++++END++++++++++
Please enter a number !
1
Apr 14, 2024 10:26:51 PM org.hibernate.jpa.ir
INFO: HHH000204: Processing PersistenceUnitIr
Apr 14, 2024 10:26:52 PM org.hibernate.Versic
INFO: HHH000412: Hibernate ORM core version 5
Apr 14, 2024 10:26:52 PM org.hibernate.annota
INFO: HCANN000001: Hibernate Commons Annotati
Apr 14, 2024 10:26:52 PM org.hibernate.engine
WARN: HHH10001002: Using Hibernate built-in c
Apr 14, 2024 10:26:53 PM org.hibernate.engine
INFO: HHH10001005: using driver [com.microsof
Apr 14, 2024 10:26:53 PM org.hibernate.engine
INFO: HHH10001001: Connection properties: {pa
Apr 14, 2024 10:26:53 PM org.hibernate.engine
INFO: HHH10001003: Autocommit mode: false
Apr 14, 2024 10:26:53 PM org.hibernate.engine
INFO: HHH000115: Hibernate connection pool si
Apr 14, 2024 10:26:53 PM org.hibernate.dialec
INFO: HHH000400: Using dialect: org.hibernate
Apr 14, 2024 10:26:54 PM org.hibernate.resour
INFO: HHH10001501: Connection obtained from J
```

25. Result


SQLQuery2.sql - localhost.tutorial (sa (68)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
tutorial
Execute
Object Explorer
Connect
localhost (SQL Server 16.0.1000.6 - sa)
Databases
System Databases
Database Snapshots
CandidateManagement
Equipments2024DB
tutorial
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.hibernate_sequence
dbo.STUDENTS
Dropped Ledger Tables
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
Security
Server Objects
Replication
Always On High Availability
Management
Integration Services Catalogs
SQLQuery2.sql - loca...st.tutorial (sa (68))
SQLQuery1.sql - loc...nts2024DB (sa (57))*
SELECT TOP (1000) [id]
,[fistName]
,[lastName]
,[marks]
FROM [tutorial].[dbo].[STUDENTS]
100 %
Results
Messages
id fistName lastName marks
1 1 Lam Nguyen 9
Query executed successfully.
localhost (16.0 RTM) sa (68) tutorial 00:00:00 1 rows
Properties

FPT Education
FPT UNIVERSITY

Java

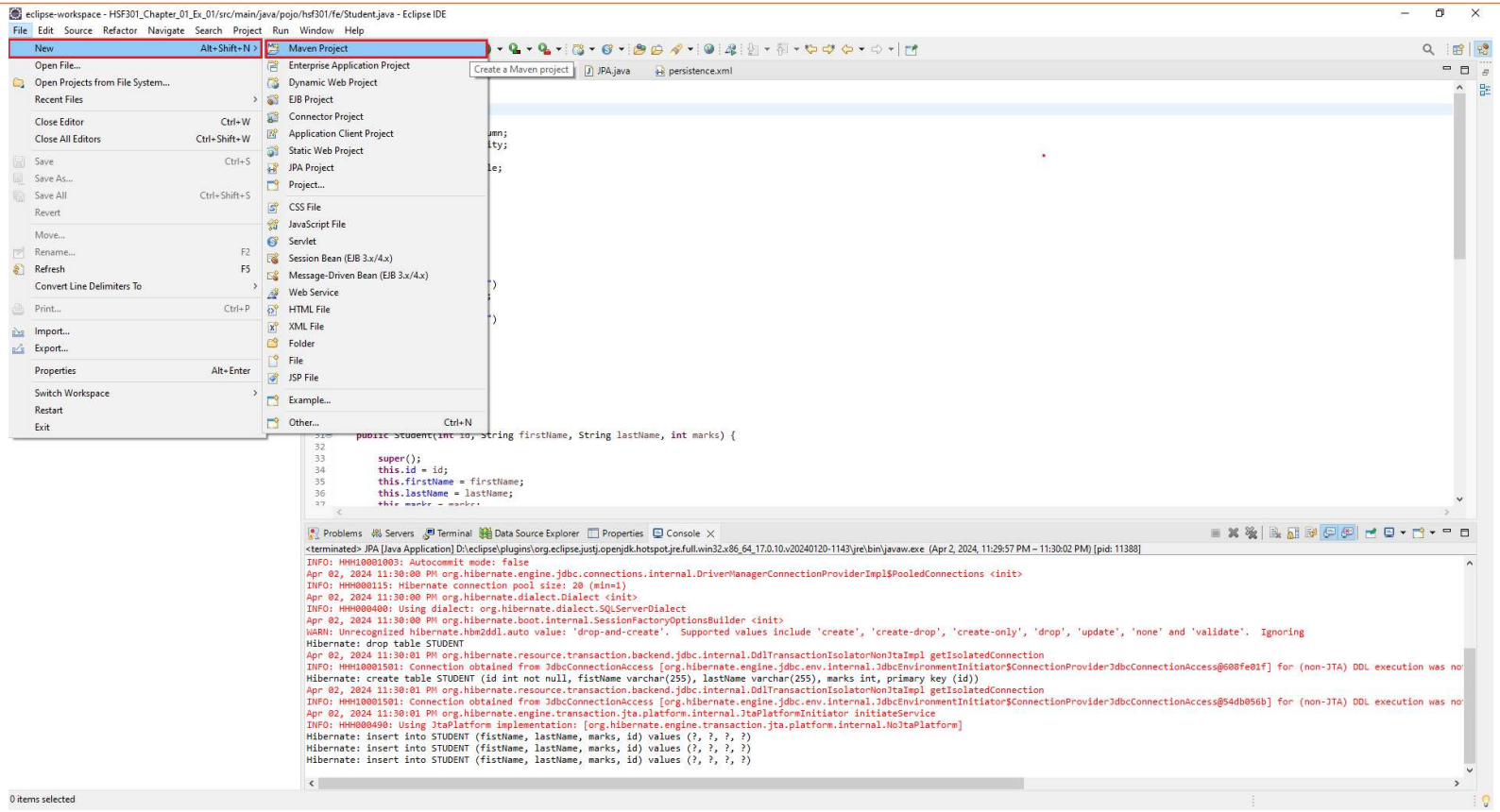
Mapping in JPA

Relationships Annotations in JPA

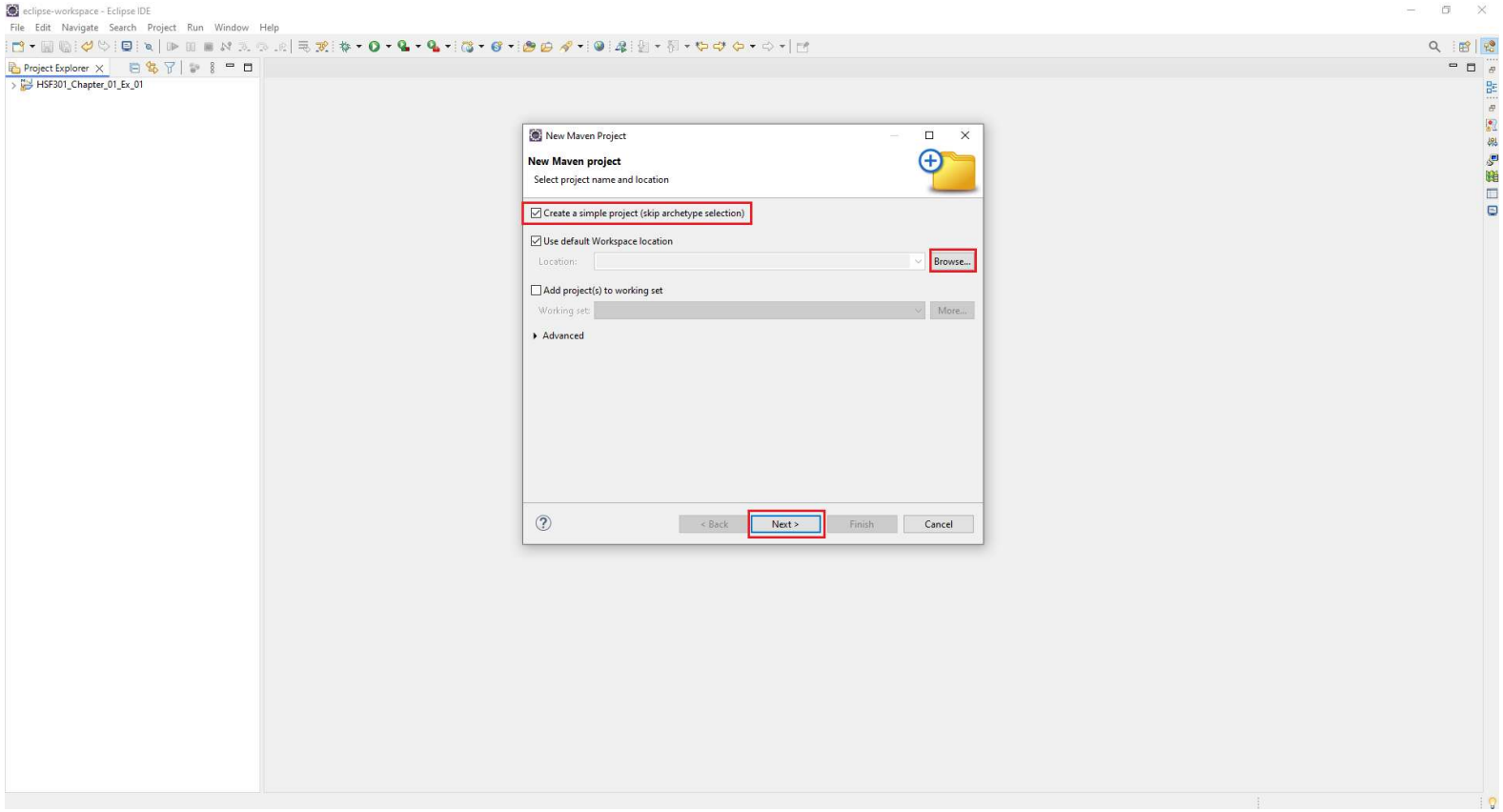
- **@ManyToOne:** This annotation defines a many-to-one relationship between two entities.
- **@OneToMany:** This annotation defines a one-to-many relationship between two entities.
- **@OneToOne:** This annotation defines a one-to-one relationship between two entities.
- **@ManyToMany:** This annotation defines a many-to-many relationship between two entities.

Demo JPA
(One To Many)

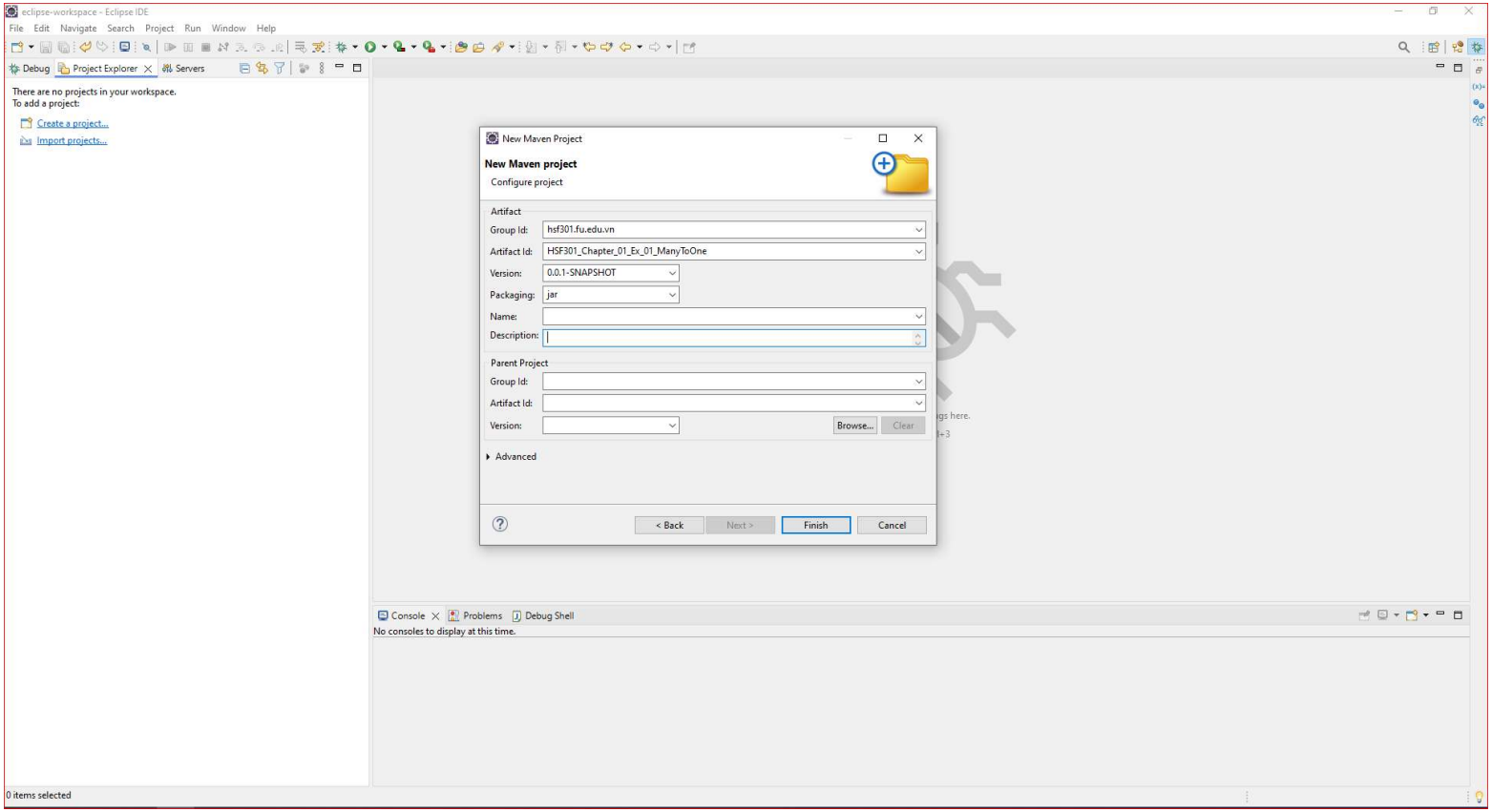
1. Open Eclipse, File | New | Maven Project



2. Check Create a simple project -> Browse Project -> Next



3. Fill the information Project -> Click Finish



4. Structure of Maven Project

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/pom.xml - Eclipse IDE

File Edit Source Navigate Search Project Run Design Window Help

Project Explorer | Servers

- HSF301_Chapter_01_Ex_01_ManyToOne [HSF301 main]
  - JPA Content
    - persistence.xml
  - src/main/java
    - hsf301.fe.dao
    - hsf301.fe.gui
    - hsf301.fe.pojo
    - hsf301.fe.repository
    - hsf301.fe.service
  - src/main/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

HSF301_Chapter_01_Ex_01_ManyToOne/pom.xml

```
https://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation)
1  <project xmlns="http://maven.apache.org/POM/4.0.0"
2      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd":
4      <modelVersion>4.0.0</modelVersion>
5      <groupId>fe.edu.vn</groupId>
6      <artifactId>HSF301_Chapter_01_Ex_01_ManyToOne</artifactId>
7      <version>0.0.1-SNAPSHOT</version>
8      <dependencies>
9          <dependency>
10             <groupId>org.eclipse.persistence</groupId>
11             <artifactId>javax.persistence</artifactId>
12             <version>2.2.0</version>
13         </dependency>
14         <dependency>
15             <groupId>org.hibernate</groupId>
16             <artifactId>hibernate-entitymanager</artifactId>
17             <version>5.6.9.Final</version>
18         </dependency>
19         <dependency>
20             <groupId>com.microsoft.sqlserver</groupId>
21             <artifactId>mssql-jdbc</artifactId>
22             <version>11.2.3.jre17</version>
23         </dependency>
24     </dependencies>
25 </project>
```

Overview | Dependencies | Dependency Hierarchy | Effective POM | pom.xml

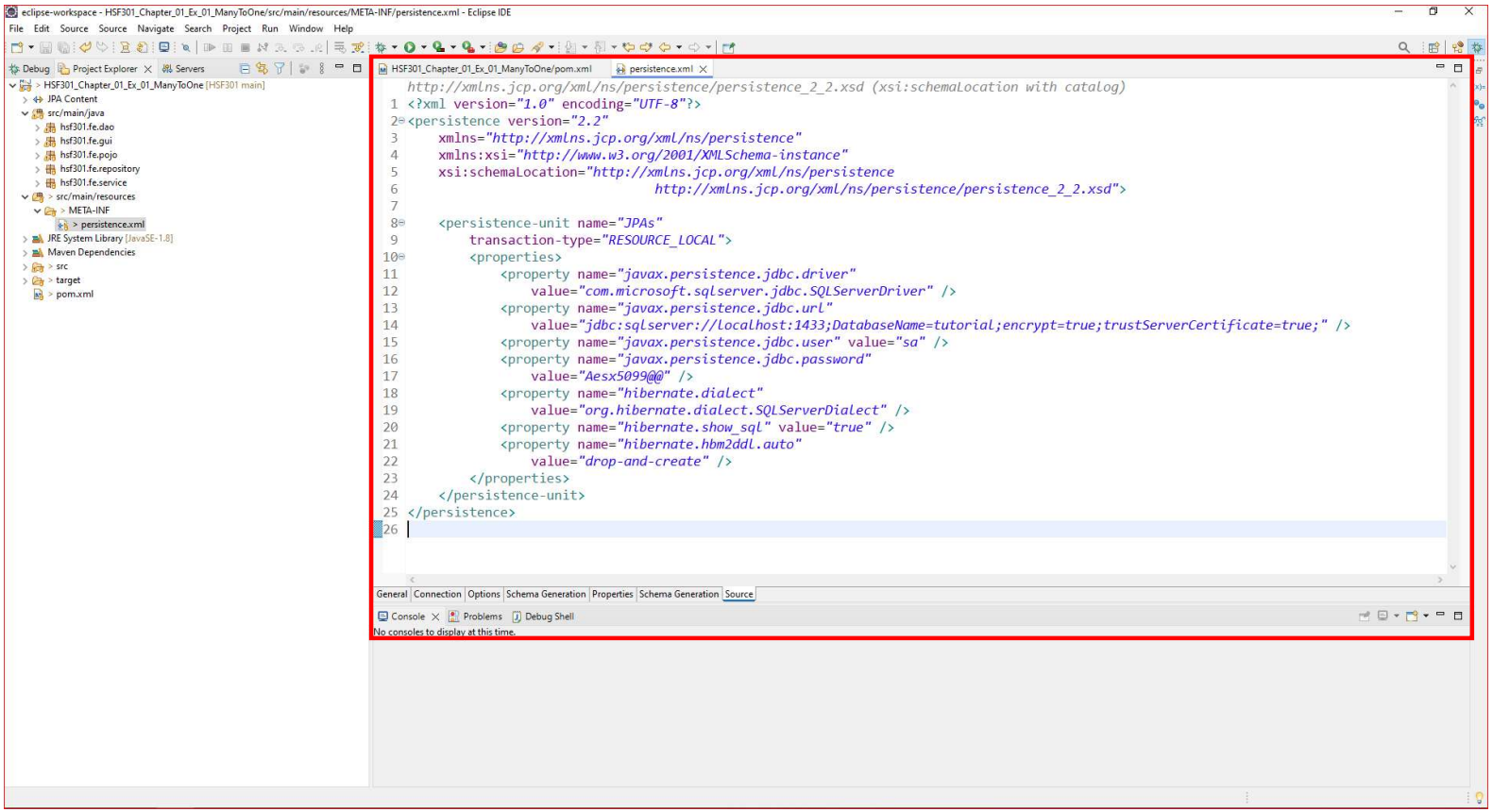
Console | Problems | Debug Shell

No consoles to display at this time.

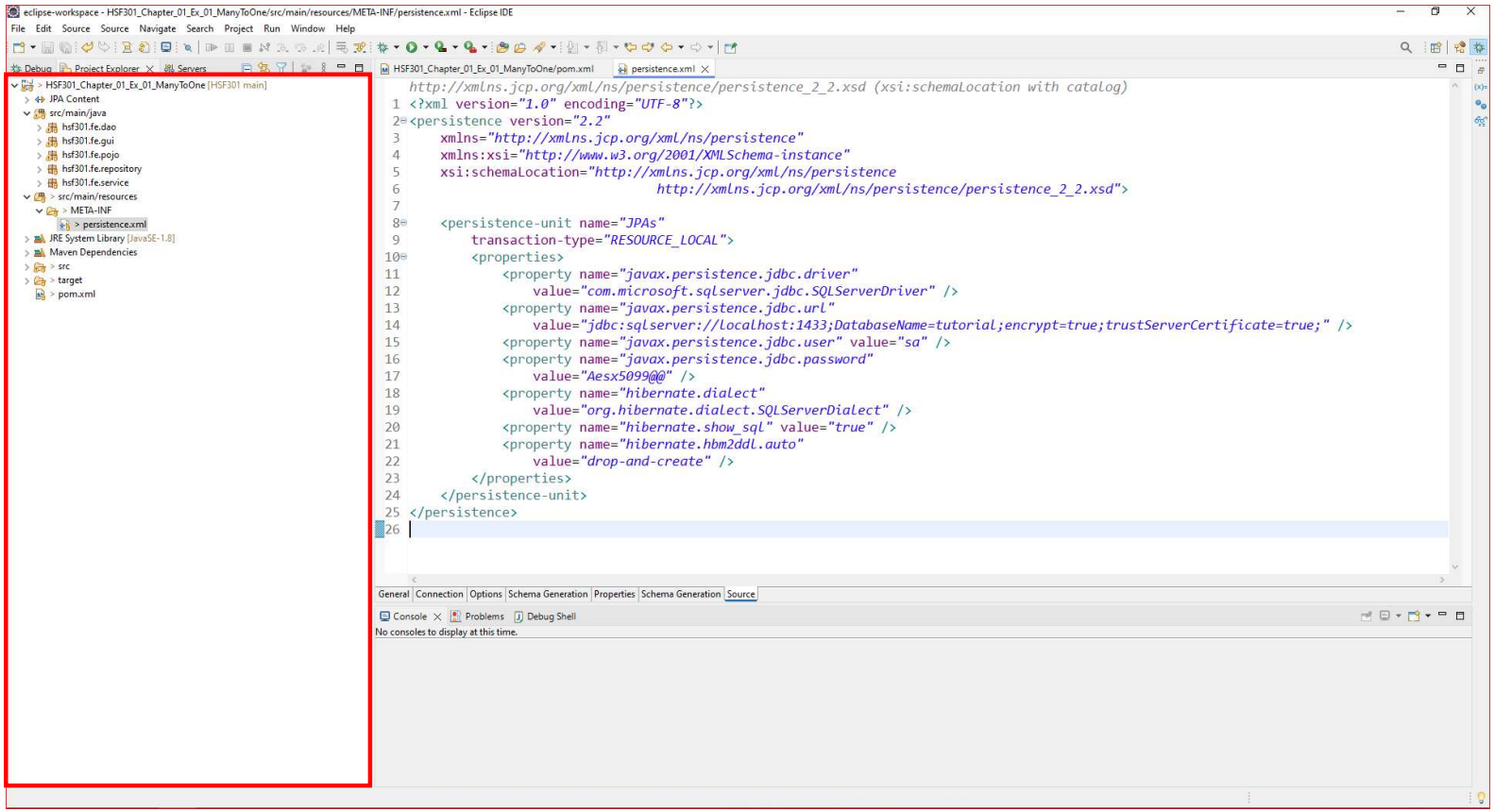
Writable | Insert | 12 : 37 : 530

Edit the content of pom.xml

5. Create persistence.xml in META-INF folder



6. Create structure of project



7. Create Books in Pojo

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/src/main/java/hsf301/fe/pojo/Book.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer | Servers

- HSF301_Chapter_01_Ex_01_ManyToOne [HSF301 main
  - JPA Content
    - persistence.xml
  - src/main/java
    - hsf301.fe.dao
    - hsf301.fe.gui
    - hsf301.fe.pojo
      - Book.java
      - Student.java
    - hsf301.fe.repository
    - hsf301.fe.service
  - src/main/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Book.java

```java
package hsf301.fe.pojo;

import javax.persistence.CascadeType;

@Entity
@Table(name = "BOOKS")
public class Book {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    @Column(name = "title", length = 30)
    private String title;

    private String author;

    private String isbn;

    @ManyToOne(cascade = CascadeType.ALL)
    @JoinColumn(name = "student_id")
    private Student student;

    public Book() {
        super();
    }

}
```

Console | Problems | Debug Shell

No consoles to display at this time.

Writable | Smart Insert | 28 : 42 : 646

8. Create Students in Pojo

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/src/main/java/hsf301/fe/pojo/Student.java - Eclipse IDE

```java
package hsf301.fe.pojo;

import java.util.HashSet;

@Entity
@Table(name = "STUDENTS")
public class Student {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private int id;

    @Column(name = "fistName", nullable = false, unique = true)
    private String firstName;

    @Column(name = "lastName")
    private String lastName;

    @Column(name = "marks")
    private int marks;

    @OneToMany(cascade = CascadeType.ALL)
    @JoinColumn(name = "student_id")
    private Set<Book> books;

    public Student() {
```

9. Create StudentDAO

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/src/main/java/hsf301/fe/dao/StudentDAO.java - Eclipse IDE

```
1  package hsf301.fe.dao;
2
3⊕ import java.util.List;
10
11 public class StudentDAO {
12
13     private static EntityManager em;
14     private static EntityManagerFactory emf;
15
16⊕    public StudentDAO(String persistanceName) {
19
20⊕    public void save(Student student) {
35
36⊕    public List<Student> getStudents() {
51
52⊕    public void delete(Long studentID) {
65
66⊕    public Student findById(Long studentID) {
80
81⊕    public void update(Student student) {
98 }
99
```

10. Save Student in StudentDAO

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/src/main/java/hsf301/fe/dao/StudentDAO.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer | Servers

- HSF301_Chapter_01_Ex_01_ManyToOne [HSF301 main
  - JPA Content
    - persistence.xml
  - src/main/java
    - hsf301.fe.dao
      - StudentDAO.java
    - hsf301.fe.gui
    - hsf301.fe.pojo
    - hsf301.fe.repository
    - hsf301.fe.service
  - src/main/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Book.java | Student.java | *StudentDAO.java

```java
import java.util.List;

public class StudentDAO {

    private static EntityManager em;
    private static EntityManagerFactory emf;

    public StudentDAO(String persistanceName) {
        emf = Persistence.createEntityManagerFactory(persistanceName);
    }

    public void save(Student student) {
        try {
            em = emf.createEntityManager();
            em.getTransaction().begin();
            em.merge(student);
            em.getTransaction().commit();
        } catch (Exception ex) {
            em.getTransaction().rollback();
            System.out.println("Error " + ex.getMessage());
        } finally {
            em.close();

        }
    }
```

Console | Problems | Debug Shell

No consoles to display at this time.

Writable | Smart Insert | 34 : 1 : 774

11. Get All Students in StudentDAO

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/src/main/java/hsf301/fe/dao/StudentDAO.java - Eclipse IDE

```java
    private static EntityManager em;
    private static EntityManagerFactory emf;

    public StudentDAO(String persistanceName) {
        emf = Persistence.createEntityManagerFactory(persistanceName);
    }

    public void save(Student student) {

    public List<Student> getStudents() {
        List<Student> students = null;
        try {
            em = emf.createEntityManager();
            em.getTransaction().begin();
            students = em.createQuery("from Student").getResultList();
        } catch (Exception ex) {

            System.out.println("Error " + ex.getMessage());
        } finally {
            em.close();
        }
        return students;

    }

    public void delete(Long studentID) {
```

FPT Education
FPT UNIVERSITY

12. Delete Student in StudentDAO

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/src/main/java/hsf301/fe/dao/StudentDAO.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer | Servers

- HSF301_Chapter_01_Ex_01_ManyToOne [HSF301 main
  - JPA Content
    - persistence.xml
  - src/main/java
    - hsf301.fe.dao
      - StudentDAO.java
    - hsf301.fe.gui
    - hsf301.fe.pojo
    - hsf301.fe.repository
    - hsf301.fe.service
  - src/main/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Book.java | Student.java | *StudentDAO.java

```java
public class StudentDAO {

    private static EntityManager em;
    private static EntityManagerFactory emf;

    public StudentDAO(String persistanceName) {
        emf = Persistence.createEntityManagerFactory(persistanceName);
    }

    public void save(Student student) {

    public List<Student> getStudents() {

    public void delete(Long studentID) {
        try {
            em = emf.createEntityManager();
            em.getTransaction().begin();
            Student s = em.find(Student.class, studentID);
            em.remove(s);
            em.getTransaction().commit();
        } catch (Exception ex) {
            System.out.println("Error " + ex.getMessage());
        } finally {
            em.close();
        }
    }
```

Console | Problems | Debug Shell

No consoles to display at this time.

Writable | Smart Insert | 34 : 1 : 774

FPT Education
FPT UNIVERSITY

13. Get a Student in StudentDAO

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/src/main/java/hsf301/fe/dao/StudentDAO.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer | Servers

- HSF301_Chapter_01_Ex_01_ManyToOne [HSF301 main
  - JPA Content
    - persistence.xml
  - src/main/java
    - hsf301.fe.dao
      - StudentDAO.java
    - hsf301.fe.gui
    - hsf301.fe.pojo
    - hsf301.fe.repository
    - hsf301.fe.service
  - src/main/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Book.java | Student.java | *StudentDAO.java

```
16      public StudentDAO(String persistanceName) {
17          emf = Persistence.createEntityManagerFactory(persistanceName);
18      }
19
20      public void save(Student student) {
34
35      public List<Student> getStudents() {
50
51      public void delete(Long studentID) {
64
65      public Student findById(Long studentID) {
66          Student student = null;
67          try {
68              em = emf.createEntityManager();
69              em.getTransaction().begin();
70              student = em.find(Student.class, studentID);
71          } catch (Exception ex) {
72              System.out.println("Error " + ex.getMessage());
73          } finally {
74              em.close();
75          }
76          return student;
77
78      }
79
80      public void update(Student student) {
97  }
```

Console | Problems | Debug Shell

No consoles to display at this time.

Writable | Smart Insert | 34 : 1 : 774

FPT Education
FPT UNIVERSITY

14. Get a Student in StudentDAO

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/src/main/java/hsf301/fe/dao/StudentDAO.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer | Servers

- HSF301_Chapter_01_Ex_01_ManyToOne [HSF301 main
  - JPA Content
    - persistence.xml
  - src/main/java
    - hsf301.fe.dao
      - StudentDAO.java
    - hsf301.fe.gui
    - hsf301.fe.pojo
    - hsf301.fe.repository
    - hsf301.fe.service
  - src/main/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Book.java | Student.java | *StudentDAO.java

```
20⊕     public void save(Student student) {
34
35⊕     public List<Student> getStudents() {
50
51⊕     public void delete(Long studentID) {
64
65⊕     public Student findById(Long studentID) {
79
80⊖     public void update(Student student) {
81          try {
82              em = emf.createEntityManager();
83              em.getTransaction().begin();
84              Student s = em.find(Student.class, student.getId());
85              if (s != null) {
86                  s.setFirstName(student.getFirstName());
87                  s.setLastName(student.getLastName());
88                  s.setMarks(student.getMarks());
89                  em.getTransaction().commit();
90              }
91          } catch (Exception ex) {
92              System.out.println("Error " + ex.getMessage());
93          } finally {
94              em.close();
95          }
96      }
97  }
98
```

Console | Problems | Debug Shell

No consoles to display at this time.

Writable | Smart Insert | 34 : 1 : 774

15. Create IStudentRepository

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/src/main/java/hsf301/fe/repository/IStudentRepository.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer | Servers

- HSF301_Chapter_01_Ex_01_ManyToOne [HSF301 main
  - JPA Content
    - persistence.xml
  - src/main/java
    - hsf301.fe.dao
    - hsf301.fe.gui
    - hsf301.fe.pojo
    - hsf301.fe.repository
      - IStudentRepository.java
      - StudentRepository.java
    - hsf301.fe.service
  - src/main/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

IStudentRepository.java

```java
package hsf301.fe.repository;

import java.util.List;

public interface IStudentRepository {
    public List<Student> findAll();

    public void save(Student student);

    public void delete(Long studentID);

    public Student findById(Long studentID);

    public void update(Student student);
}
```

Console | Problems | Debug Shell

No consoles to display at this time.

Writable | Smart Insert | 1 : 1 : 0

16. Create StudentRepository

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/src/main/java/hsf301/fe/repository/StudentRepository.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer | Servers

- HSF301_Chapter_01_Ex_01_ManyToOne [HSF301 main
  - JPA Content
    - persistence.xml
  - src/main/java
    - hsf301.fe.dao
    - hsf301.fe.gui
    - hsf301.fe.pojo
    - hsf301.fe.repository
      - IStudentRepository.java
      - StudentRepository.java
    - hsf301.fe.service
  - src/main/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

IStudentRepository.java | StudentRepository.java

```java
package hsf301.fe.repository;

import java.util.List;

public class StudentRepository implements IStudentRepository {
    private StudentDAO studentDAO = null;

    public StudentRepository(String fileConfig) {
        studentDAO = new StudentDAO(fileConfig);
    }
    @Override
    public void save(Student student) {
        // TODO Auto-generated method stub
        studentDAO.save(student);
    }
    @Override
    public List<Student> findAll() {
        // TODO Auto-generated method stub
        return studentDAO.getStudents();
    }
    @Override
    public void delete(Long studentID) {
        studentDAO.delete(studentID);
    }
    @Override
    public Student findById(Long studentID) {
        // TODO Auto-generated method stub
        return studentDAO.findById(studentID);
    }
    @Override
    public void update(Student student) {
        studentDAO.update(student);
    }
```

Console | Problems | Debug Shell

No consoles to display at this time.

Writable | Smart Insert | 35 : 36 : 885

17. Create IStudentService

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/src/main/java/hsf301/fe/service/IStudentService.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer | Servers

- HSF301_Chapter_01_Ex_01_ManyToOne [HSF301 main
  - JPA Content
    - persistence.xml
  - src/main/java
    - hsf301.fe.dao
    - hsf301.fe.gui
    - hsf301.fe.pojo
    - hsf301.fe.repository
    - hsf301.fe.service
      - IStudentService.java
      - StudentService.java
  - src/main/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

IStudentRepository.java | StudentRepository.java | IStudentService.java

```java
package hsf301.fe.service;

import java.util.List;

public interface IStudentService {
    public List<Student> findAll();

    public void save(Student student);

    public void delete(Long studentID);

    public Student findById(Long studentID);

    public void update(Student student);

}
```

Console | Problems | Debug Shell

No consoles to display at this time.

Writable | Smart Insert | 1 : 1 : 0

FPT Education
FPT UNIVERSITY

18. Create StudentService

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/src/main/java/hsf301/fe/service/StudentService.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer | Servers

- HSF301_Chapter_01_Ex_01_ManyToOne [HSF301 main
  - JPA Content
    - persistence.xml
  - src/main/java
    - hsf301.fe.dao
    - hsf301.fe.gui
    - hsf301.fe.pojo
    - hsf301.fe.repository
    - hsf301.fe.service
      - IStudentService.java
      - StudentService.java
  - src/main/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

IStudentRepository.java | StudentRepository.java | IStudentService.java | *StudentService.java

```java
import java.util.List;
public class StudentService implements IStudentService {
    private IStudentRepository iStudentRepo = null;
    public StudentService(String fileName) {
        iStudentRepo = new StudentRepository(fileName);
    }
    @Override
    public void save(Student student) {
        iStudentRepo.save(student);
    }
    @Override
    public List<Student> findAll() {
        return iStudentRepo.findAll();
    }
    @Override
    public void delete(Long studentID) {
        iStudentRepo.delete(studentID);
    }
    @Override
    public Student findById(Long studentID) {
        return iStudentRepo.findById(studentID);
    }
    @Override
    public void update(Student student) {
        iStudentRepo.update(student);
    }
}
```

Console | Problems | Debug Shell

No consoles to display at this time.

Writable | Smart Insert | 26 : 49 : 747

19. Create Main function

```java
package hsf301.fe.gui;

import java.util.ArrayList;

public class ManyToOne {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        //Many To
        String fileName = "JPAs";
        IStudentService studentService = new StudentService(fileName);
        Student student = new Student("Lam","Nguyen",9);
        Book book = new Book("Java Persistence with Spring","Catalin Tudose","9781617299186");
        student.getBooks().add(book);
        studentService.AddStudent(student);
    }

}
```

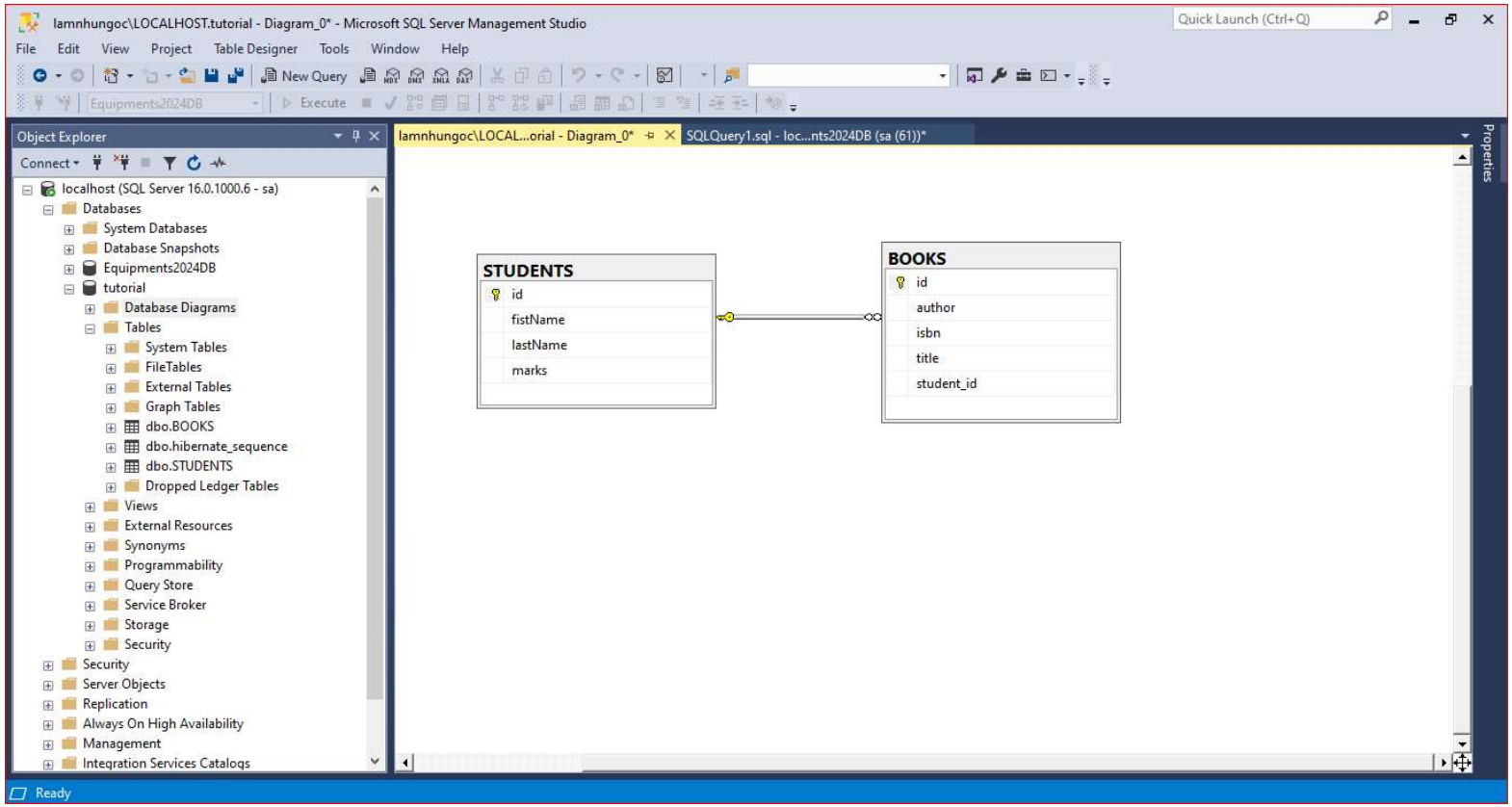
20. Run Program

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToOne/src/main/java/hsf301/fe/gui/ManyToOne.java - Eclipse IDE

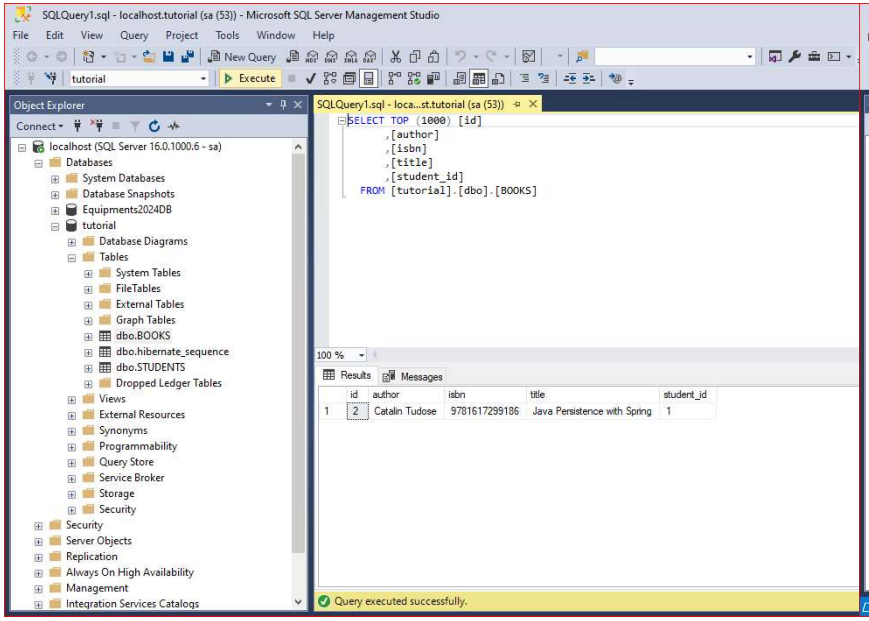
```java
package hsf301.fe.gui;

import hsf301.fe.pojo.Book;

public class ManyToOne {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        //Many To
        String fileName = "JPAs";
        IStudentService studentService = new StudentService(fileName);
        Student student = new Student("Lam","Nguyen",9);
        Book book = new Book("Java Persistence with Spring","Catalin Tudose","9781617299186");
        student.getBooks().add(book);
        studentService.save(student);
    }

}
```

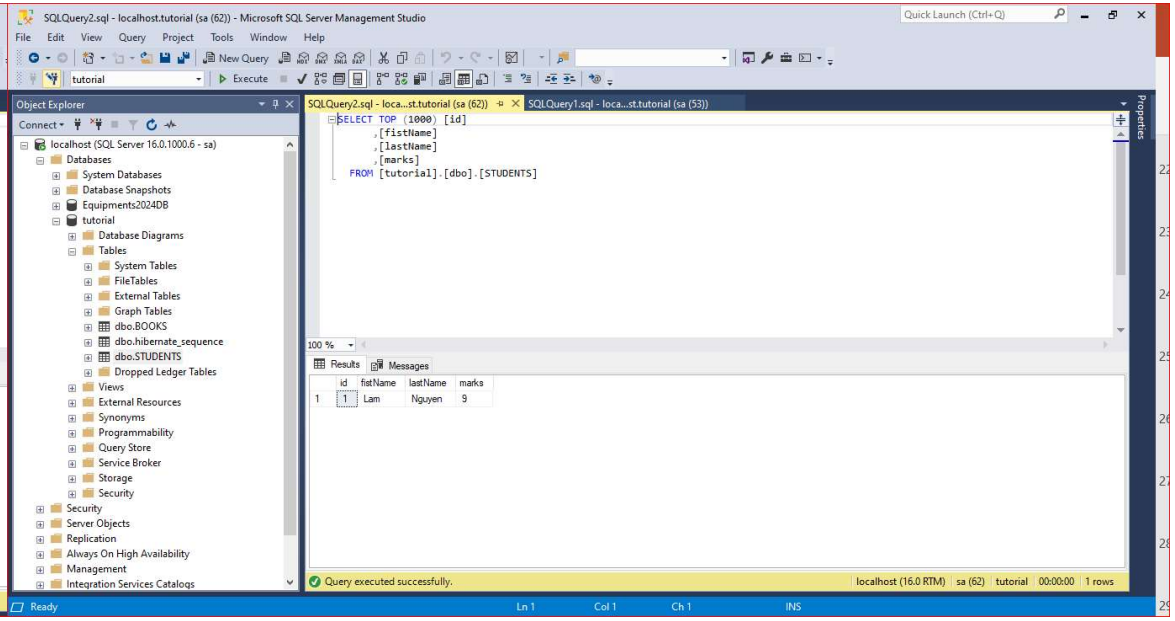
Non-externalized string literal; it should be followed by //$NON-NLS-<n>$ | Writable | Smart Insert | 17 : 55 : 496

21. Result


lamnhungoc\LOCALHOST.tutorial - Diagram_0* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Project Table Designer Tools Window Help
New Query
Equipments2024DB
Execute
Object Explorer
Connect
localhost (SQL Server 16.0.1000.6 - sa)
Databases
System Databases
Database Snapshots
Equipments2024DB
tutorial
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.BOOKS
dbo.hibernate_sequence
dbo.STUDENTS
Dropped Ledger Tables
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
Security
Server Objects
Replication
Always On High Availability
Management
Integration Services Catalogs
lamnhungoc\LOCAL...orial - Diagram_0*
SQLQuery1.sql - loc...nts2024DB (sa (61))*
Properties
STUDENTS
id
fistName
lastName
marks
BOOKS
id
author
isbn
title
student_id
Ready

22. Result


SQLQuery1.sql - localhost.tutorial (sa (53)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
New Query
tutorial
Execute
Object Explorer
Connect
localhost (SQL Server 16.0.1000.6 - sa)
Databases
System Databases
Database Snapshots
Equipments2024DB
tutorial
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.BOOKS
dbo.hibernate_sequence
dbo.STUDENTS
Dropped Ledger Tables
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
Security
Server Objects
Replication
Always On High Availability
Management
Integration Services Catalogs
SQLQuery1.sql - loca...st.tutorial (sa (53))
SELECT TOP (1000) [id]
,[author]
,[isbn]
,[title]
,[student_id]
FROM [tutorial].[dbo].[BOOKS]
100 %
Results
Messages
id author isbn title student_id
1 2 Catalin Tudose 9781617299186 Java Persistence with Spring 1
Query executed successfully.


SQLQuery2.sql - localhost.tutorial (sa (62)) - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
tutorial
Execute
Object Explorer
Connect
localhost (SQL Server 16.0.1000.6 - sa)
Databases
System Databases
Database Snapshots
Equipments2024DB
tutorial
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.BOOKS
dbo.hibernate_sequence
dbo.STUDENTS
Dropped Ledger Tables
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
Security
Server Objects
Replication
Always On High Availability
Management
Integration Services Catalogs
SQLQuery2.sql - loca...st.tutorial (sa (62))
SQLQuery1.sql - loca...st.tutorial (sa (53))
SELECT TOP (1000) [id]
,[fistName]
,[lastName]
,[marks]
FROM [tutorial].[dbo].[STUDENTS]
Properties
100 %
Results
Messages
id fistName lastName marks
1 1 Lam Nguyen 9
Query executed successfully.
localhost (16.0 RTM) sa (62) tutorial 00:00:00 1 rows
Ready
Ln 1 Col 1 Ch 1 INS

Demo JPA
(Many To Many)

1. Create Books in Pojo's Package

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToMany/src/main/java/hsf301/fe/pojo/Book.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer | Servers

- HSF301_Chapter_01_Ex_01_ManyToMany [HSF301 main]
  - JPA Content
    - persistence.xml
  - src/main/java
    - hsf301.fe.dao
      - StudentDAO.java
    - hsf301.fe.gui
      - ManyToManyDemo.java
    - hsf301.fe.pojo
      - Book.java
      - Student.java
    - hsf301.fe.repository
    - hsf301.fe.service
  - src/main/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

persistence.xml | StudentDAO.java | Book.java

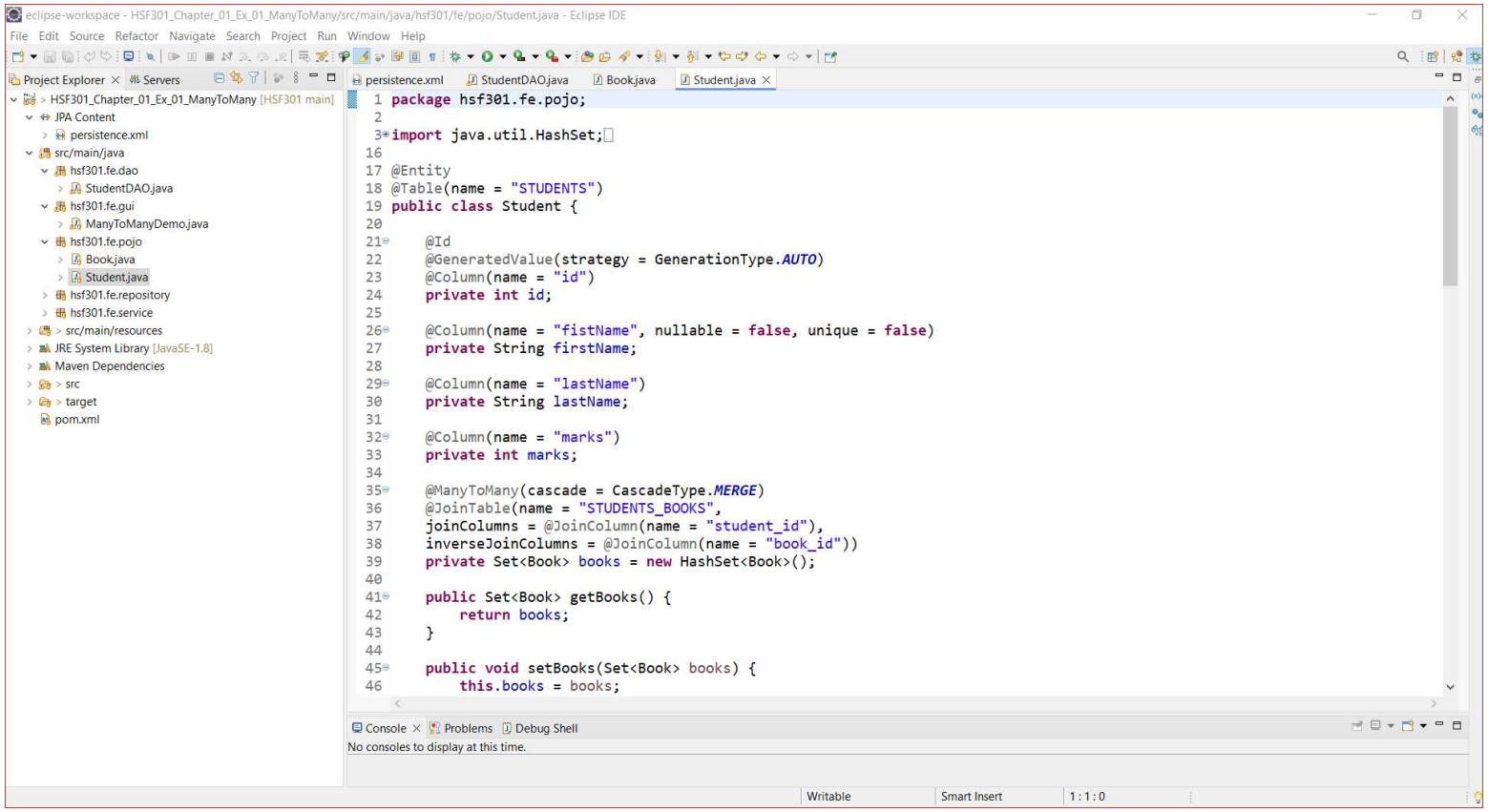
```java
package hsf301.fe.pojo;

import java.util.HashSet;

@Entity
@Table(name = "BOOKS")

public class Book {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "id")
    private Long id;

    @Column(name = "title", length = 30)
    private String title;

    private String author;

    private String isbn;

    @ManyToMany(mappedBy = "books")
    private Set<Student> student = new HashSet<Student>();

    public Book() {
        super();
    }
    public Book(String title, String author, String isbn) {
        super();
        this.title = title;
        this.author = author;
        this.isbn = isbn;
    }

```

Console | Problems | Debug Shell

No consoles to display at this time.

Writable | Smart Insert | 1 : 1 : 0

2. Create Students in Pojo's Package



3. Run Program

eclipse-workspace - HSF301_Chapter_01_Ex_01_ManyToMany/src/main/java/hsf301/fe/gui/ManyToManyDemo.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer | Servers

- HSF301_Chapter_01_Ex_01_ManyToMany [HSF301 main]
  - JPA Content
    - persistence.xml
  - src/main/java
    - hsf301.fe.dao
      - StudentDAO.java
    - hsf301.fe.gui
      - ManyToManyDemo.java
    - hsf301.fe.pojo
      - Book.java
      - Student.java
    - hsf301.fe.repository
    - hsf301.fe.service
  - src/main/resources
  - JRE System Library [JavaSE-1.8]
  - Maven Dependencies
  - src
  - target
  - pom.xml

persistence.xml | StudentDAO.java | Book.java | Student.java | ManyToManyDemo.java

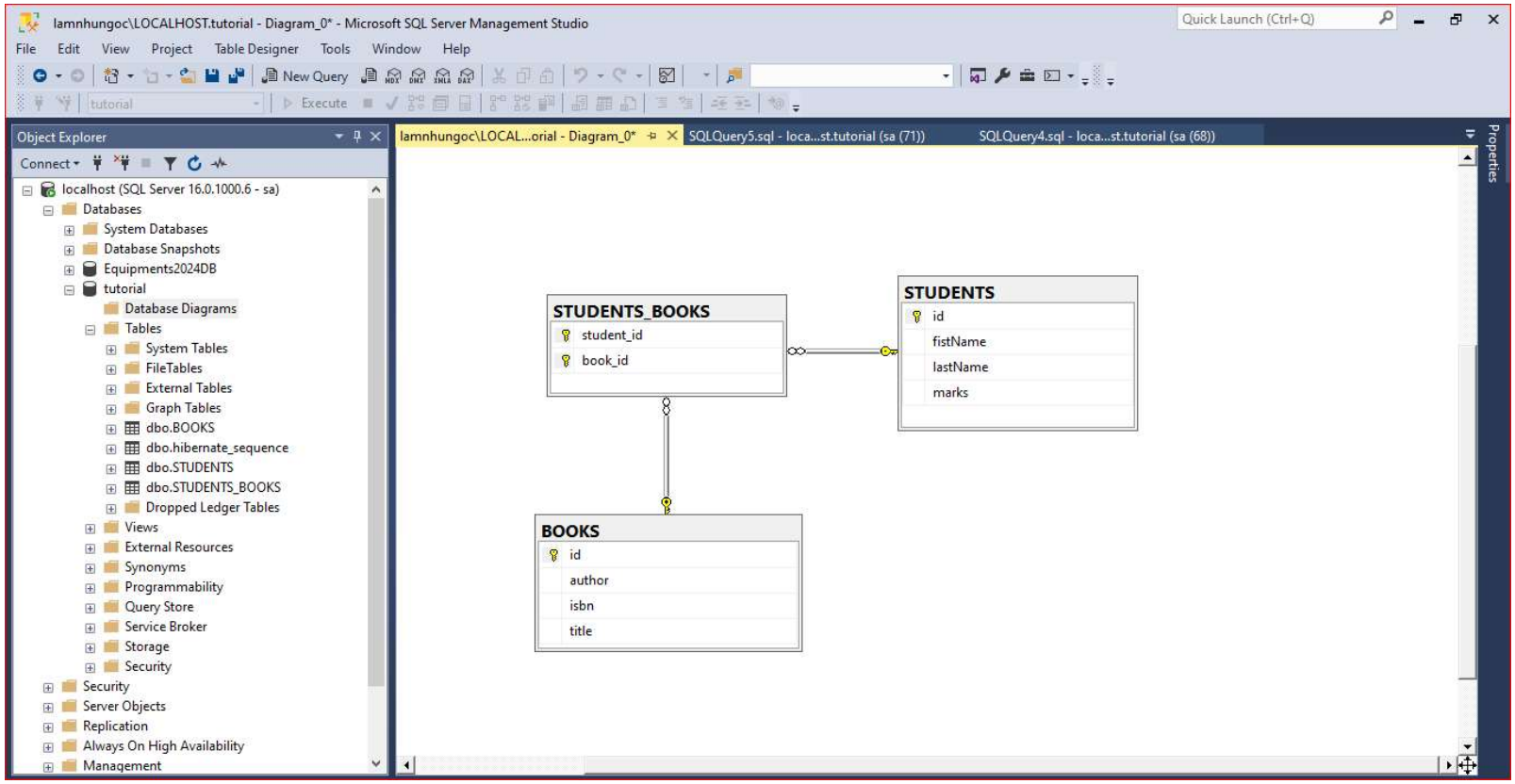
```java
package hsf301.fe.gui;

import java.util.List;

public class ManyToManyDemo {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        String fileName = "JPAs";
        IStudentService studentService = new StudentService(fileName);
        Student student = new Student("Lam", "Nguyen", 9);
        Book book = new Book("Java Persistence with Spring", "Catalin Tudose", "9781617299186");
        student.getBooks().add(book);
        studentService.save(student);
        List<Student> students = studentService.findAll();
        for(Student st : students) {
            System.out.println(st.getFirstName());
        }

    }
}
```

Console | Problems | Debug Shell

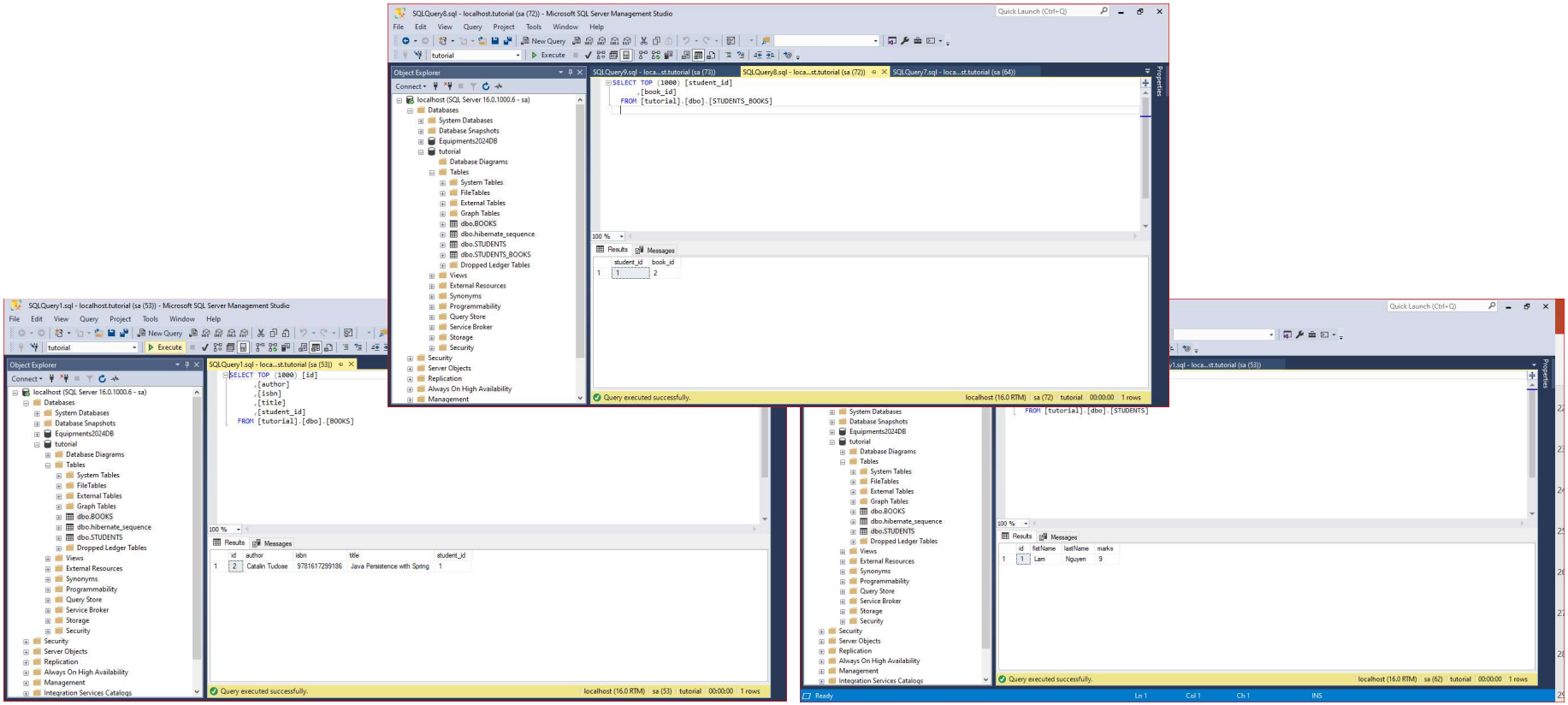
```
<terminated> ManyToManyDemo [Java Application] D:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-1143\jre\bin\javaw.exe (Apr 7, 2024, 10:36:10 PM – 10:36:14 PM) [pid: 4608
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: select next_val as id_val from hibernate_sequence with (updlock, rowlock)
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: insert into STUDENTS (fistName, lastName, marks, id) values (?, ?, ?, ?)
Hibernate: insert into BOOKS (author, isbn, title, id) values (?, ?, ?, ?)
Hibernate: insert into STUDENTS_BOOKS (student_id, book_id) values (?, ?)
Hibernate: select student0_.id as id1_1_, student0_.fistName as fistname2_1_, student0_.lastName as lastname3_1_, student0_.marks
Lam
```

Non-externalized string literal; it should be followed by //$NON-NLS-<n>$ | Writable | Smart Insert | 18 : 39 : 501

4. Result


lamnhungoc\LOCALHOST.tutorial - Diagram_0* - Microsoft SQL Server Management Studio
File Edit View Project Table Designer Tools Window Help
New Query
Execute
Object Explorer
Connect
localhost (SQL Server 16.0.1000.6 - sa)
Databases
System Databases
Database Snapshots
Equipments2024DB
tutorial
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.BOOKS
dbo.hibernate_sequence
dbo.STUDENTS
dbo.STUDENTS_BOOKS
Dropped Ledger Tables
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
Security
Server Objects
Replication
Always On High Availability
Management
lamnhungoc\LOCAL...orial - Diagram_0*
SQLQuery5.sql - loca...st.tutorial (sa (71))
SQLQuery4.sql - loca...st.tutorial (sa (68))
STUDENTS_BOOKS
student_id
book_id
STUDENTS
id
fistName
lastName
marks
BOOKS
id
author
isbn
title

5. Result


SQLQuery8.sql - localhost.tutorial (sa (72)) - Microsoft SQL Server Management Studio
SELECT TOP (1000) [student_id]
,[book_id]
FROM [tutorial].[dbo].[STUDENTS_BOOKS]
Query executed successfully.
SQLQuery1.sql - localhost.tutorial (sa (53)) - Microsoft SQL Server Management Studio
SELECT TOP (1000) [id]
,[author]
,[isbn]
,[title]
,[student_id]
FROM [tutorial].[dbo].[BOOKS]
Query executed successfully.
SELECT TOP (1000) [id]
,[fistName]
,[lastName]
,[marks]
FROM [tutorial].[dbo].[STUDENTS]
Query executed successfully.

Summary

- Concepts were introduced:
  - Overview about JPA
  - Architecture Overview new features of JPA
  - Why JPA is selected as develop application?
  - Explain and demo using Eclipse IDE to create JPA Console App
  - Create and Run cross-platform Console application with Java connect to MSSQL with Repository Pattern