

Jakarta Persistence API

<https://jakarta.ee/specifications/persistence/3.0/jakarta-persistence-spec-3.0>

Phần 1: Object-relational mapping (ORM)

Cài đặt MariaDB

<https://mariadb.org/>

[Download MariaDB Server](#)
REST API

Release Schedule

Reporting Bugs

MariaDB Server Statistics

Download MariaDB Server Documentation

View all releases for:
MariaDB Server
Connector/C
Connector/J
Connector/ODBC
Connector/Python
Connector/Node.js

MariaDB Server Version
MariaDB Server 11.2.2

Display older releases: ☐

Operating System
Windows

Architecture
x86_64

Package Type
MSI Package

Download

Mirror
BKNS.VN - Hanoi

Release date: 2023-11-21
File name: mariadb-11.2.2-winx64.msi
File size: 72.4 MB
• Download galera-26.4.16
Display signature and checksums

MariaDB 11.2 (x64) Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

MariaDB Server

Database instance

Client Programs

Backup utilities

Development Components

Third party tools

HeidiSQL

Installs C/C++ header files and libraries

This feature requires 11MB on your hard drive. It has 1 of 1 subfeatures selected. The subfeatures require 396KB on your hard drive.

Location: C:\MariaDB 11.2\ Browse...

Reset Disk Usage Back Next Cancel

User settings

Default instance properties
MariaDB 11.2 (x64) database configuration

☒ **Modify password for database user 'root'**

New root password: Enter new root password

Confirm: Retype the password

☒ **Enable access from remote machines for 'root' user**

☒ **Use UTF8 as default server's character set**

Data directory:

User name: root

Password: root

Database settings

Default instance properties
MariaDB 11.2 (x64) database configuration

☒ **Install as service**

Service Name:

☒ **Enable networking**

TCP port:

InnoDB engine settings

Buffer pool size: MB

Page size: KB

HeidiSQL

- HeidiSQL là một công cụ quản trị cơ sở dữ liệu quan hệ mã nguồn mở.
- Nó hỗ trợ nhiều cơ sở dữ liệu quan hệ như MySQL, PostgreSQL, MariaDB, MSSQL, SQLite, Oracle, DB2, SQL Server, Sybase, ...
- Cho phép lưu trữ, cập nhật, xóa dữ liệu từ cơ sở dữ liệu quan hệ.

Filter ...

Session name ^	Host
mysession	127.0...

Settings Advanced SSL Statistics

Network type: MariaDB or MySQL (TCP/IP)

Library: libmariadb.dll

Hostname / IP: 127.0.0.1

☐ Prompt for credentials

☐ Use Windows authentication

User: root

Password:

Port: 3306

☐ Compressed client/server protocol

Databases: Separated by semicolon

Comment:

+ New Save Delete Open Cancel More

mysession\ - HeidiSQL 12.3.0.6589

File Edit Search Query Tools Go to Help

Database filter Table filter

Host: 127.0.0.1 Query

Databases (4) Variables Status Processes Command-Statistics

Database ^	Size	Items	Last ...	Tables	Views	Func...	Proc...	Trigg...	Events
information_schema									
mysql	3.3 MiB	31	2024...	30	1	0	0	0	0
performance_schema									
sys									

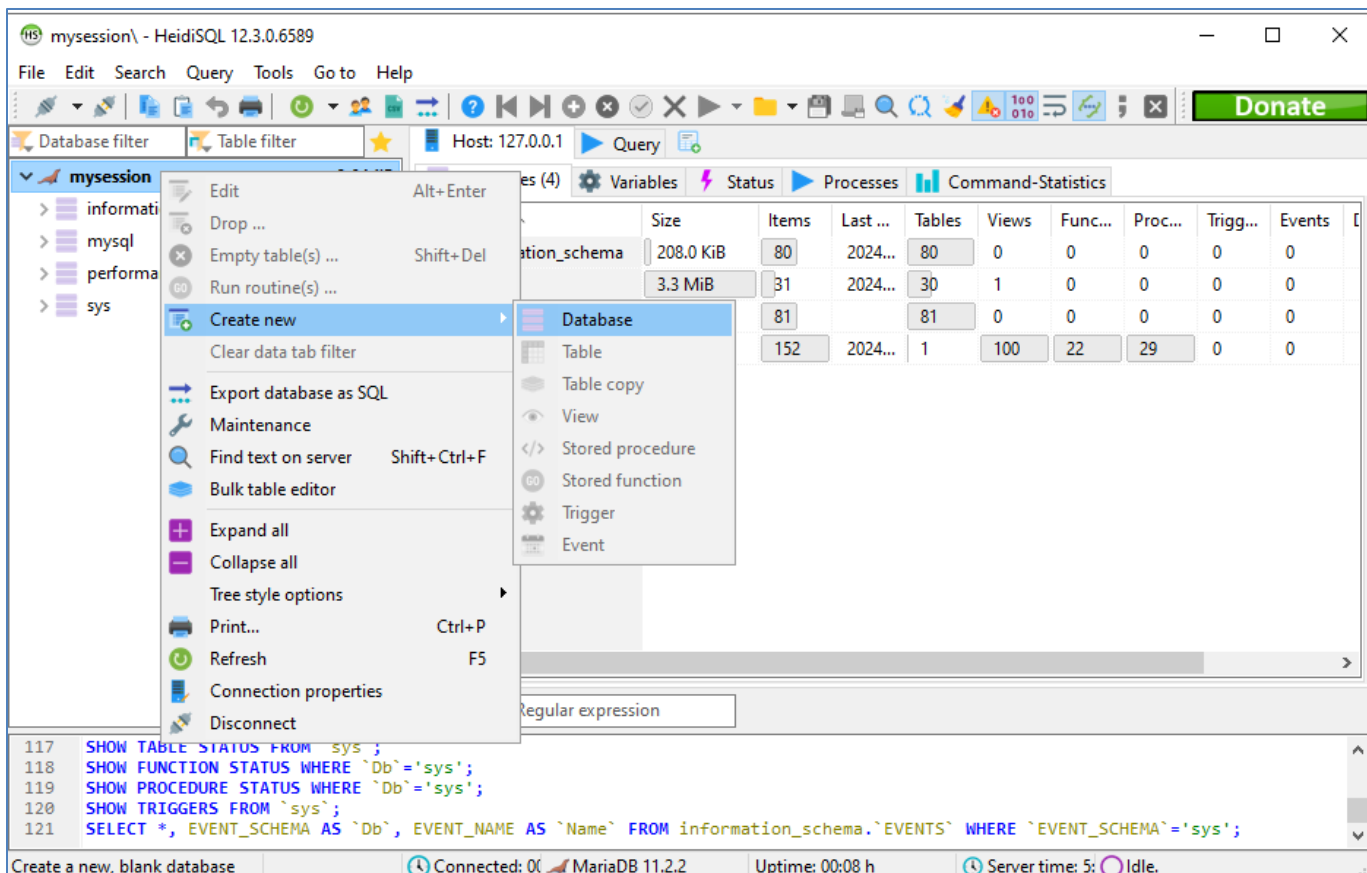
Filter: Regular expression

```

27 SHOW TABLE STATUS FROM `mysql`;
28 SHOW FUNCTION STATUS WHERE `Db`='mysql';
29 SHOW PROCEDURE STATUS WHERE `Db`='mysql';
30 SHOW TRIGGERS FROM `mysql`;
31 SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='mysql';

```

Connected: 0x MariaDB 11.2.2 Uptime: 00:05 h Server time: 5: Idle.



Jakarta Persistence API (JPA)

- JPA - ORM là gì?

- JPA là viết tắt của Java Persistence API
- ORM là viết tắt của Object-Relational Mapping
- Là một phần của Java EE và được sử dụng để ánh xạ mô hình đối tượng của Java với cơ sở dữ liệu quan hệ.
- JPA cho phép ánh các lớp thực thể của Java vào các bảng trong cơ sở dữ liệu quan hệ.
- Cho phép ánh xạ các quan hệ giữa các lớp thực thể của Java thành các quan hệ giữa các bảng trong cơ sở dữ liệu quan hệ.
- Nó cung cấp một cách tiêu chuẩn để tạo, lưu trữ, cập nhật và xóa dữ liệu từ cơ sở dữ liệu quan hệ sử dụng các lớp thực thể của Java.

- Anatomy of an Entity - Cấu trúc của một thực thể

- Một entity là POJO (Plain Old Java Object) được đánh dấu bằng @Entity annotation.
- Một lớp thực thể biểu diễn cho một bảng trong cơ sở dữ liệu quan hệ.
- Bắt buộc phải có một constructor không tham số (*default constructor*).
- Bắt buộc phải có một hoặc vài fields được đánh dấu bằng @Id annotation để biểu diễn là primary key trong bảng.
 - Khóa chính có một field gọi là khóa đơn (*Single primary key*)

- Khóa chính có từ hai fields tham gia vào khóa gọi là khóa phức hợp (*Composite primary key*)
- Lớp thực thể không thể là final class hoặc enum.
- Những field được đánh dấu bằng @Transient annotation sẽ không được ánh xạ vào cơ sở dữ liệu.
- Mỗi đối tượng được tạo từ một lớp thực thể, được biểu diễn cho một hàng trong bảng.
- Mỗi field của một lớp thực thể, được biểu diễn là một cột trong bảng.
- **Deployment tools**
 - Java: Java 1.8+
 - RDBMS: MySQL, PostgreSQL, MariaDB, H2, MSSQL ...
 - NoSQL: Neo4j
 - Tools for Java developers: IntelliJ IDEA, Eclipse IDE for Java EE Developers
 - Framework/Library: Hibernate / eclipselink, Neo4j OGM Driver
- **Eclipselink là gì?**
 - EclipseLink là một framework ORM (*Object-Relational Mapping*) mã nguồn mở được phát triển bởi Eclipse Foundation.
 - Nó cung cấp một cách tiêu chuẩn để ánh xạ mô hình đối tượng của Java với cơ sở dữ liệu quan hệ.
 - Nó cung cấp một cách tiêu chuẩn để tạo, lưu trữ, cập nhật và xóa dữ liệu từ cơ sở dữ liệu quan hệ sử dụng các lớp thực thể của Java.
- **Hibernate ORM là gì?**
 - Hibernate ORM là một framework ORM (*Object-Relational Mapping*) mã nguồn mở được phát triển bởi Red Hat.
 - Nó cung cấp một cách tiêu chuẩn để ánh xạ mô hình đối tượng của Java với cơ sở dữ liệu quan hệ.
 - Nó cung cấp một cách tiêu chuẩn để tạo, lưu trữ, cập nhật và xóa dữ liệu từ cơ sở dữ liệu quan hệ sử dụng các lớp thực thể của Java.
- **Ưu điểm:**
 - Hỗ trợ nhiều cơ sở dữ liệu quan hệ như: MySQL, PostgreSQL, MariaDB, H2, MSSQL, Oracle...
 - Hỗ trợ nhiều tính năng như: lazy loading, caching, inheritance, validation, logging, querying, sorting, filtering, JPQL, criteria, ...

Tạo JPA project

Tạo một Maven Project → Click phải trên project → Properties → Project Facets → (*May be: Convert to faceted from...*) → check vào JPA → Further configuration available ... → JPA Implementation: Disable Library Configuration → OK → Apply and Close

Hoặc tạo JPA Project, sau đó Convert to Maven Project

Installation with EclipseLink and (MariaDB or MSSQL)

<https://eclipse.dev/eclipselink/documentation/4.0/solutions/solutions.html#CHDDDDAB>

Maven project

```
<dependencies>
  <!--
  https://mvnrepository.com/artifact/org.eclipse.persistence/eclipselink -->
  <dependency>
    <groupId>org.eclipse.persistence</groupId>
    <artifactId>eclipselink</artifactId>
    <version>4.0.2</version>
  </dependency>
  <!--
  https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-api -->
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-api</artifactId>
    <version>5.10.2</version>
    <scope>test</scope>
  </dependency>
  <!--
  https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-engine -->
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-engine</artifactId>
    <version>5.10.2</version>
    <scope>test</scope>
  </dependency>
  <!--
  https://mvnrepository.com/artifact/com.microsoft.sqlserver/mssql-jdbc -->
  <dependency>
    <groupId>com.microsoft.sqlserver</groupId>
    <artifactId>mssql-jdbc</artifactId>
    <version>12.3.0.jre20-preview</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.30</version>
    <scope>provided</scope>
```

```

</dependency>
<!--
https://mvnrepository.com/artifact/org.mariadb.jdbc/mariadb-java-client -->
<dependency>
    <groupId>org.mariadb.jdbc</groupId>
    <artifactId>mariadb-java-client</artifactId>
    <version>3.3.3</version>
</dependency>
</dependencies>

```

Persistence.xml

```

<persistence-unit name="JPA_ORM_Student MSSQL">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <class>entity.Student</class>
    <class>entity.Clazz</class>
    <class>entity.Course</class>
    <class>entity.Enrollment</class>
    <properties>
        <property name="jakarta.persistence.jdbc.url"
value="jdbc:sqlserver://localhost:1433;databaseName=StudentDB;trustServerCertificate=
true" />
        <property name="jakarta.persistence.jdbc.user" value="sa" />
        <property name="jakarta.persistence.jdbc.password"
value="sapassword" />
        <property name="jakarta.persistence.jdbc.driver"
value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
        <property name="jakarta.persistence.jdbc.dialect"
value="org.hibernate.dialect.SQLServer2012Dialect" />
        <property name="hibernate.show_sql" value="true" />
        <property name="hibernate.format_sql" value="true" />
        <property name="hibernate.hbm2ddl.auto" value="update" />
    </properties>
</persistence-unit>
<persistence-unit name="JPA ORM MariaDB">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <class>entity.Student</class>
    <class>entity.Clazz</class>
    <class>entity.Course</class>
    <class>entity.Enrollment</class>

```

```

<properties>
  <property name="jakarta.persistence.jdbc.driver"
    value="org.mariadb.jdbc.Driver" />
  <property name="jakarta.persistence.jdbc.url"
    value="jdbc:mariadb://localhost:3306/testdb" />
  <property name="jakarta.persistence.jdbc.user" value="root" />
  <property name="jakarta.persistence.jdbc.password"
    value="root" />

  <property name="jakarta.persistence.jdbc.dialect"
    value="org.hibernate.dialect.MariaDBDialect" />

  <property name="hibernate.show_sql" value="true" />
  <property name="hibernate.format_sql" value="true" />
  <property name="hibernate.hbm2ddl.auto" value="update" />
</properties>
</persistence-unit>

```

Installation with Hibernate and (MariaDB or MSSQL)

https://docs.jboss.org/hibernate/stable/orm/userguide/html_single/Hibernate_User_Guide.html

Maven project

```

<dependencies>
  <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>6.4.4.Final</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-testing -->
  <dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-testing</artifactId>
    <version>6.4.4.Final</version>
    <scope>test</scope>
  </dependency>
  <!--
  https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-api -->
  <dependency>

```



```

        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-api</artifactId>
        <version>5.10.2</version>
        <scope>test</scope>
    </dependency>
    <!--
    https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-engine -->
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-engine</artifactId>
        <version>5.10.2</version>
        <scope>test</scope>
    </dependency>

    <!--
    https://mvnrepository.com/artifact/com.microsoft.sqlserver/mssql-jdbc -->
    <dependency>
        <groupId>com.microsoft.sqlserver</groupId>
        <artifactId>mssql-jdbc</artifactId>
        <version>12.3.0.jre20-preview</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <version>1.18.30</version>
        <scope>provided</scope>
    </dependency>
    <!--
    https://mvnrepository.com/artifact/org.mariadb.jdbc/mariadb-java-client -->
    <dependency>
        <groupId>org.mariadb.jdbc</groupId>
        <artifactId>mariadb-java-client</artifactId>
        <version>3.3.3</version>
    </dependency>
</dependencies>

```

```

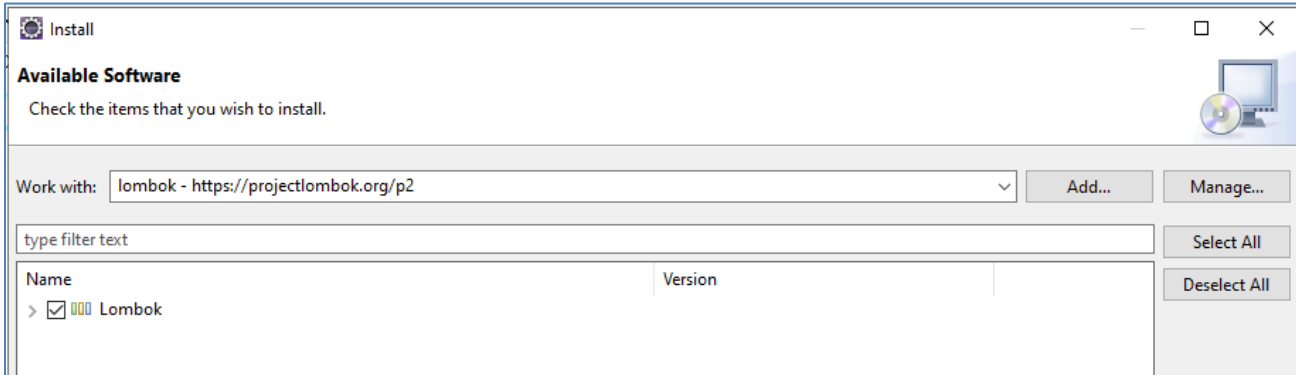
<persistence-unit name="JPA_ORM_Student MSSQL">
    <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
    <properties>
        <property name="jakarta.persistence.jdbc.url"
            value="jdbc:sqlserver://localhost:1433;databaseName=StudentDB;trustServerCertificate=
true" />
        <property name="jakarta.persistence.jdbc.user" value="sa" />
        <property name="jakarta.persistence.jdbc.password"
            value="sapassword" />
        <property name="jakarta.persistence.jdbc.driver"
            value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
        <property name="jakarta.persistence.jdbc.dialect"
            value="org.hibernate.dialect.SQLServer2012Dialect" />
        <property name="hibernate.show_sql" value="true" />
        <property name="hibernate.format_sql" value="true" />
        <property name="hibernate.hbm2ddl.auto" value="update" />
    </properties>
</persistence-unit>
<persistence-unit name="JPA ORM MariaDB">
    <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
    <properties>
        <property name="jakarta.persistence.jdbc.driver"
            value="org.mariadb.jdbc.Driver" />
        <property name="jakarta.persistence.jdbc.url"
            value="jdbc:mariadb://localhost:3306/testdb" />
        <property name="jakarta.persistence.jdbc.user" value="root" />
        <property name="jakarta.persistence.jdbc.password"
            value="root" />
        <property name="jakarta.persistence.jdbc.dialect"
            value="org.hibernate.dialect.MariaDBDialect" />
        <property name="hibernate.show_sql" value="true" />
        <property name="hibernate.format_sql" value="true" />
        <property name="hibernate.hbm2ddl.auto" value="update" />
    </properties>
</persistence-unit>

```

Cài đặt lombok trong eclipse

<https://projectlombok.org/setup/eclipse>

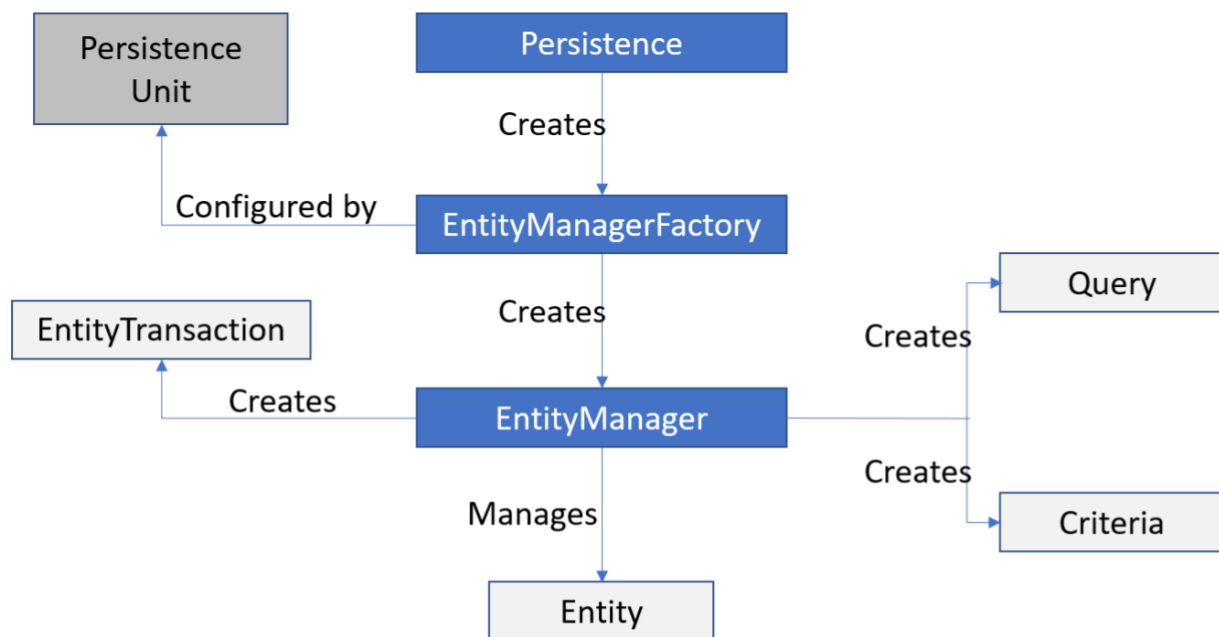
Help -> Install new Software ... ☐ Add: <https://projectlombok.org/p2>



JPA Bootstrapping

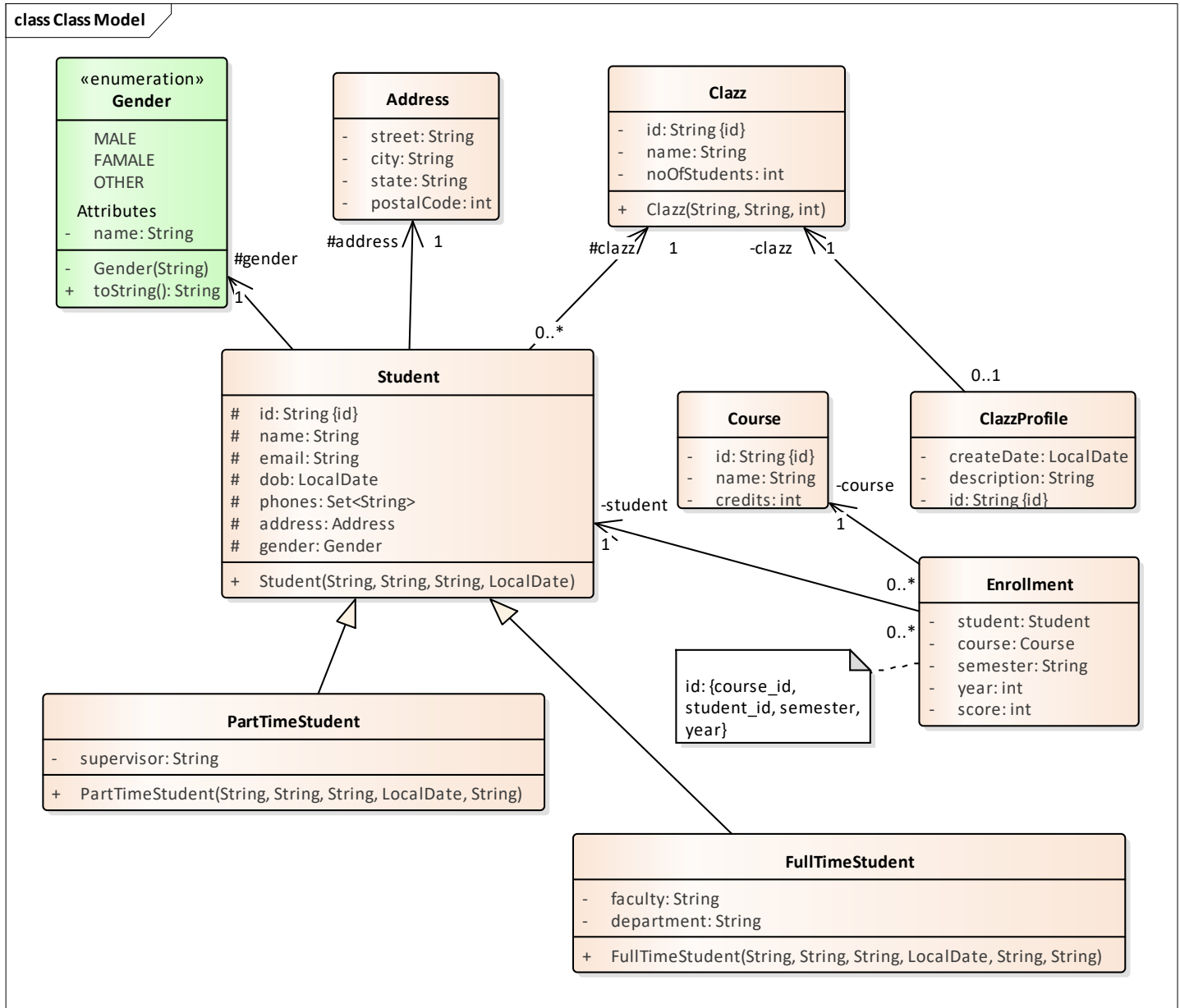
- Đặc tả JPA có 2 cách tiếp cận chính để tạo đối tượng `javax.persistence.EntityManager` từ `javax.persistence.EntityManagerFactory`, đó là: Java EE và Java SE.
- Container-bootstrapping (EE) – Container sẽ tạo `EntityManagerFactory` cho mỗi persistent-unit được cấu hình trong `META-INF/persistence.xml`.
- Application-bootstrapping (SE) - Ứng dụng tạo `javax.persistence.EntityManagerFactory` thông qua phương thức `createEntityManagerFactory` của lớp bootstrap `javax.persistence.Persistence`.
- `EntityManager` quản lý các entity, cung cấp các phương thức cho các thao tác thêm, xóa, sửa và tìm kiếm dữ liệu như `persist`, `merge`, `remove`, `find` ...

JPA Architecture



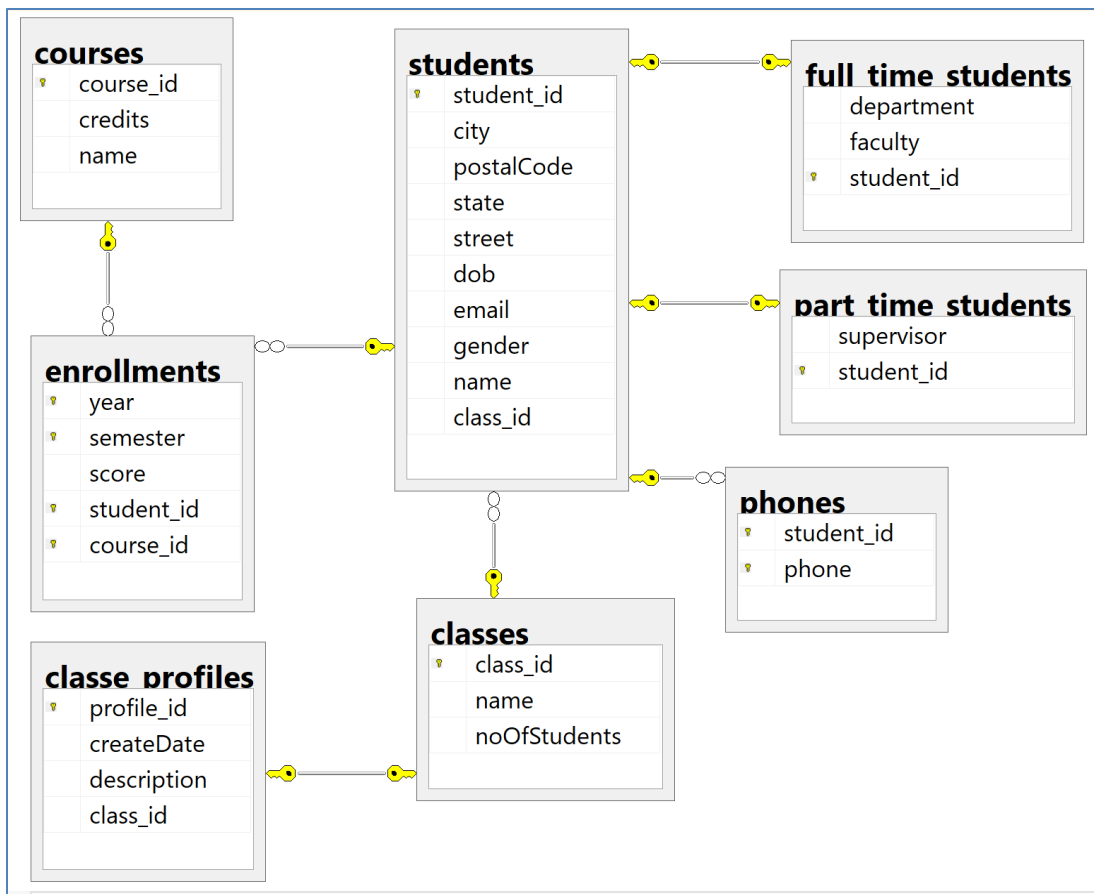
Minh họa

Cho sơ đồ lớp sau:



Yêu cầu:

- Ánh xạ mô hình lớp trên sang mô hình CSDL quan hệ tương ứng.



- (Java Programming) Viết các phương thức với các thao tác CRUD
 - a) Thực hiện các thao tác Create, Update, Delete, Find by Id trên từng đối tượng.
 - b) Tính sĩ số sinh viên theo từng lớp học, giảm dần theo sĩ số.
 - + getSisoByLophoc() : Map<Lophoc, Integer>
 - c) Tính điểm trung bình của các môn học của các sinh viên
 - + listSinhvienDiemTB() : Map<Sinhvien, Float>
 - d) Những lớp học chưa có sinh viên
 - + listLophocNull() : List<Lophoc>
 - e) Những sinh viên học môn “Lập Trình Phân tán với Công Nghệ Java” có điểm cao nhất.
 - + listSinhvienGioiJava() : List<Sinhvien>

Hướng dẫn

Lớp entity

Các lớp entity phải (*chưa xét tới mối liên kết*):

- Có default constructor,
- Khai báo @Entity và @Id,
- May be Serializable

Thuộc tính đa trị

- Cơ sở dữ liệu quan hệ sẽ tách ra thành một bảng mới, khóa chính của bảng mới là khóa chính của bảng ban đầu và thuộc tính đa trị.
- Khai báo @ElementCollection cho thuộc tính đa trị.
- Dùng @CollectionTable để mô tả thông tin cho bảng mới sẽ được tách ra.

Ví dụ:

```
@ElementCollection
@CollectionTable(name="phones", joinColumns = @JoinColumn(name="student_id"))
@Column(name="phone", nullable = false)
protected Set<String> phones;
```

Entity associations/ relationships

Mối quan hệ 1 - 1 (One to One)

- Phải xác định được đối tượng chủ thể và đối tượng phụ thuộc, cả 2 bên đều khai báo @OneToOne nếu xét 2 chiều
- Nếu xét một chiều, đi từ phía phụ thuộc sang chủ thể.
- **Trường hợp 1:** Khóa chính bên bảng phụ thuộc chính là khóa chính bên chủ thể.

Ví dụ:

```
@Table(name = "classe_profiles")
public class ClazzProfile {
    @Id
    @OneToOne
    @JoinColumn(name = "class_id")
    private Clazz clazz;
    private LocalDate createDate;
    private String description;
}
```

- **Trường hợp 2:** Bảng phía phụ thuộc có khóa chính riêng, khóa ngoại trong bảng này phải đồng thời là unique.

Ví dụ:

```
@Entity
```

```

@Table(name = "classe_profiles")
public class ClazzProfile {
    @Id
    @Column(name = "profile_id")
    private String id;
    @OneToOne
    @JoinColumn(name = "class_id", unique = true, nullable = false)
    private Clazz clazz;
    private LocalDate createDate;
    private String description;
}

```

Mối quan hệ 1 – n (*One to Many hoặc Many to One*)

- Phía One: Thêm @OneToMany với thuộc tính mappedBy.
- Phía Many thêm:
 - o @ManyToOne
 - o @JoinColumn với thuộc tính name

Nếu xét một hướng, thì hướng đi từ many to one

Ví dụ:

```

@Entity
@Table(name = "students")
public class Student {
    @Id
    @Column(name="student_id")
    protected String id;
    @Column(columnDefinition = "nvarchar(100)", nullable = false)
    protected String name;
    @Column(unique = true, nullable = false)
    protected String email;
    protected LocalDate dob;
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name="class_id")
    protected Clazz clazz;
}

```

Mối quan hệ n – n (*Many to Many*)

Nguyên tắc đối với mối quan hệ Many to Many, cơ sở dữ liệu quan hệ sẽ tách thêm một thực thể mới (2 mối quan hệ One to Many).

Khóa của thực thể kết hợp chứa ít nhất 2 khóa của 2 thực thể tham gia vào mối liên kết này.

Phải có equals và hashCode trên tất cả các thuộc tính tham gia vào khóa.

Ví dụ:

```
@EqualsAndHashCode
```

```
@Entity
```

```
@Table(name = "enrollments")
```

```
public class Enrollment {  
    @Id  
    @ManyToOne(cascade = CascadeType.ALL)  
    @JoinColumn(name = "student_id")  
    private Student student;  
    @Id  
    @ManyToOne(cascade = CascadeType.ALL)  
    @JoinColumn(name = "course_id")  
    private Course course;  
    @Id  
    private String semester;  
    @Id  
    private int year;  
  
    @EqualsAndHashCode.Exclude  
    private int score;  
}
```

Mối quan hệ n – n (không có thuộc tính của mỗi liên kết)

- Cả 2 bên điều khai báo @ManyToMany
- Bên chủ thể với thuộc tính mappedBy
- Bên còn lại thêm @JoinTable, với các thuộc tính:
 - o name = "Table_Name",
 - o joinColumns = @JoinColumn(name = "Column_name"),
 - o inverseJoinColumns = @JoinColumn(name = "Column_name")

Mối quan hệ n – n (viết lớp mô tả khóa riêng)

- Lớp thực thể mới tách thêm:
 - o Khai báo @Entity và @IdClass
 - o Id của lớp là một Composite key.
 - Gồm ít nhất 2 field là khóa của 2 lớp thực thể tham gia vào mối liên kết.
 - Mỗi field khai báo @Id, @ManyToOne và @JoinColumn
 - o Có thể có thêm các thuộc tính của mối liên kết
- Lớp mô tả composite key: các thuộc tính của lớp này là các thuộc tính tham gia vào khóa.
 - o Khai báo @Embeddable
 - o Phải có hashCode & equals,
 - o Phải có Default constructor
 - o Phải implement interface Serializable

Mối quan hệ kế thừa (*Inheritance relationship*)

Có 3 chiến lược để ánh xạ:

- InheritanceType.*SINGLE_TABLE*
- InheritanceType.*TABLE_PER_CLASS*
- InheritanceType.*JOINED*

Ví dụ:

```
@Entity
@Table(name = "students")
@Inheritance(strategy = InheritanceType.JOINED)
public class Student {
    @Id
    @Column(name="student_id")
    protected String id;
    @Column(columnDefinition = "nvarchar(100)", nullable = false)
    protected String name;
    @Column(unique = true, nullable = false)
    protected String email;
    protected LocalDate dob;
}

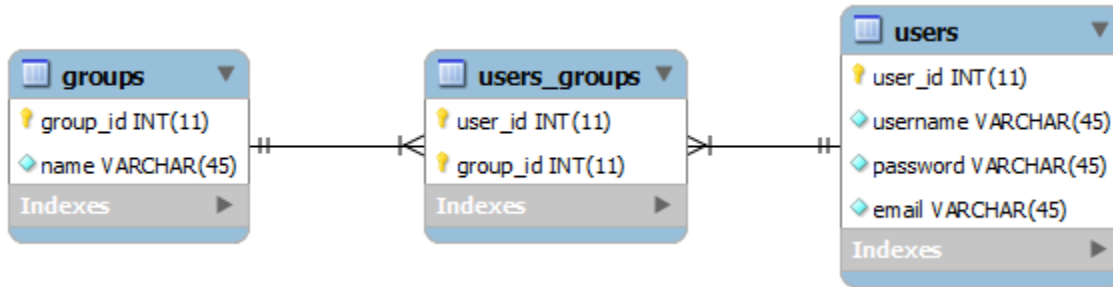
@Entity
@Table(name = "part_time_students")
public class PartTimeStudent extends Student{
    private String supervisor;
}

@Entity
@Table(name = "full_time_students")
public class FullTimeStudent extends Student{
    private String faculty;
    private String department;
}
```

Bài tập thực hành

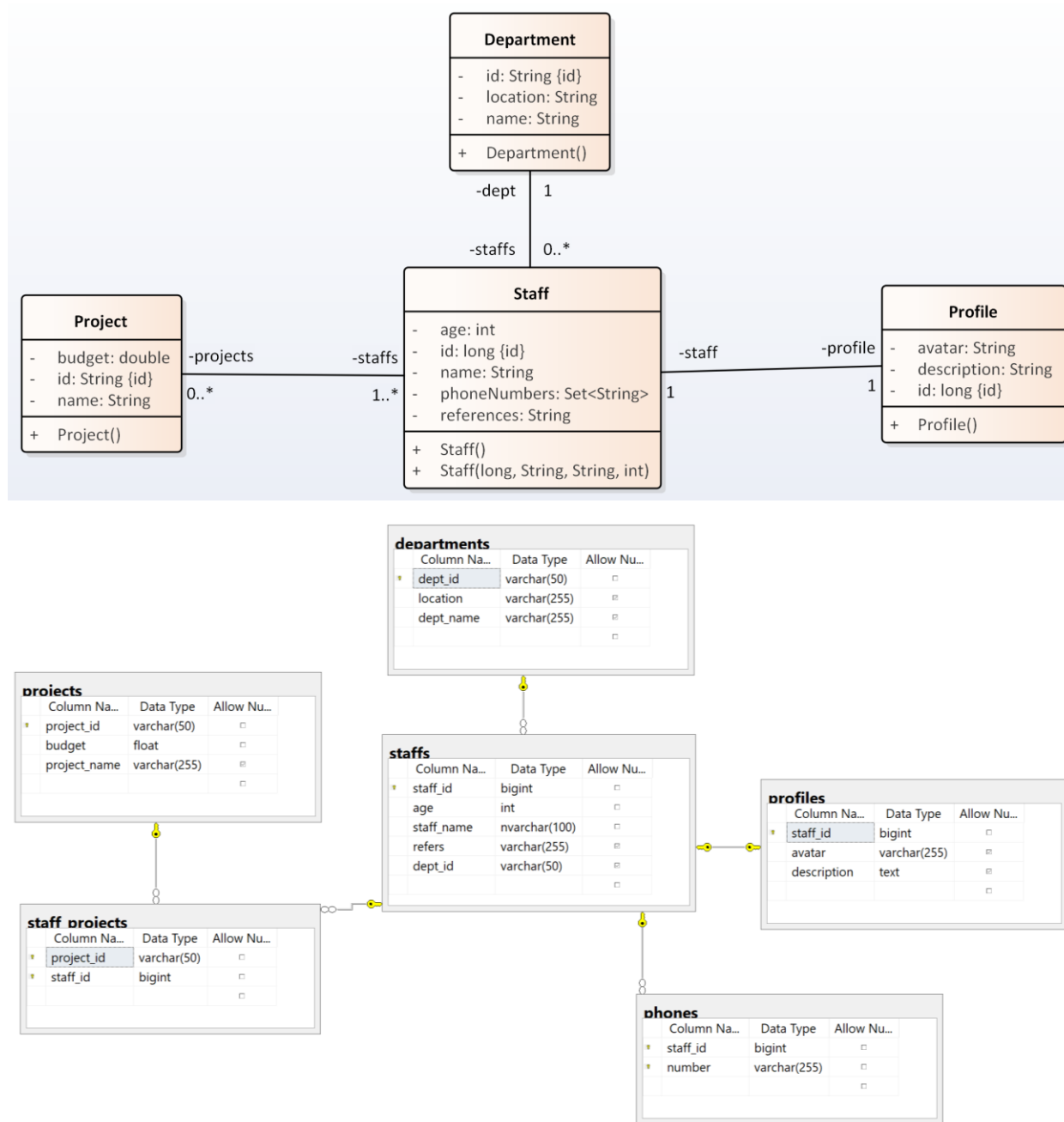
Bài tập 1

Viết các lớp persistence entity, sau cho ánh xạ ra được mô hình CSDL quan hệ sau:



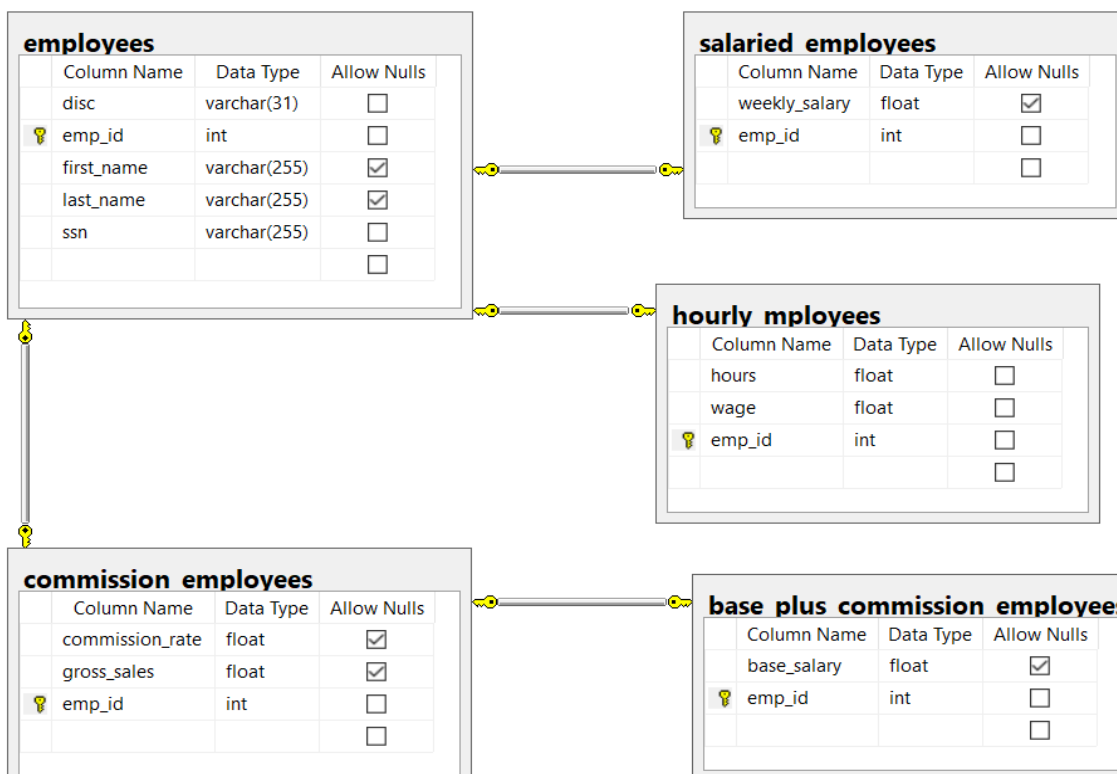
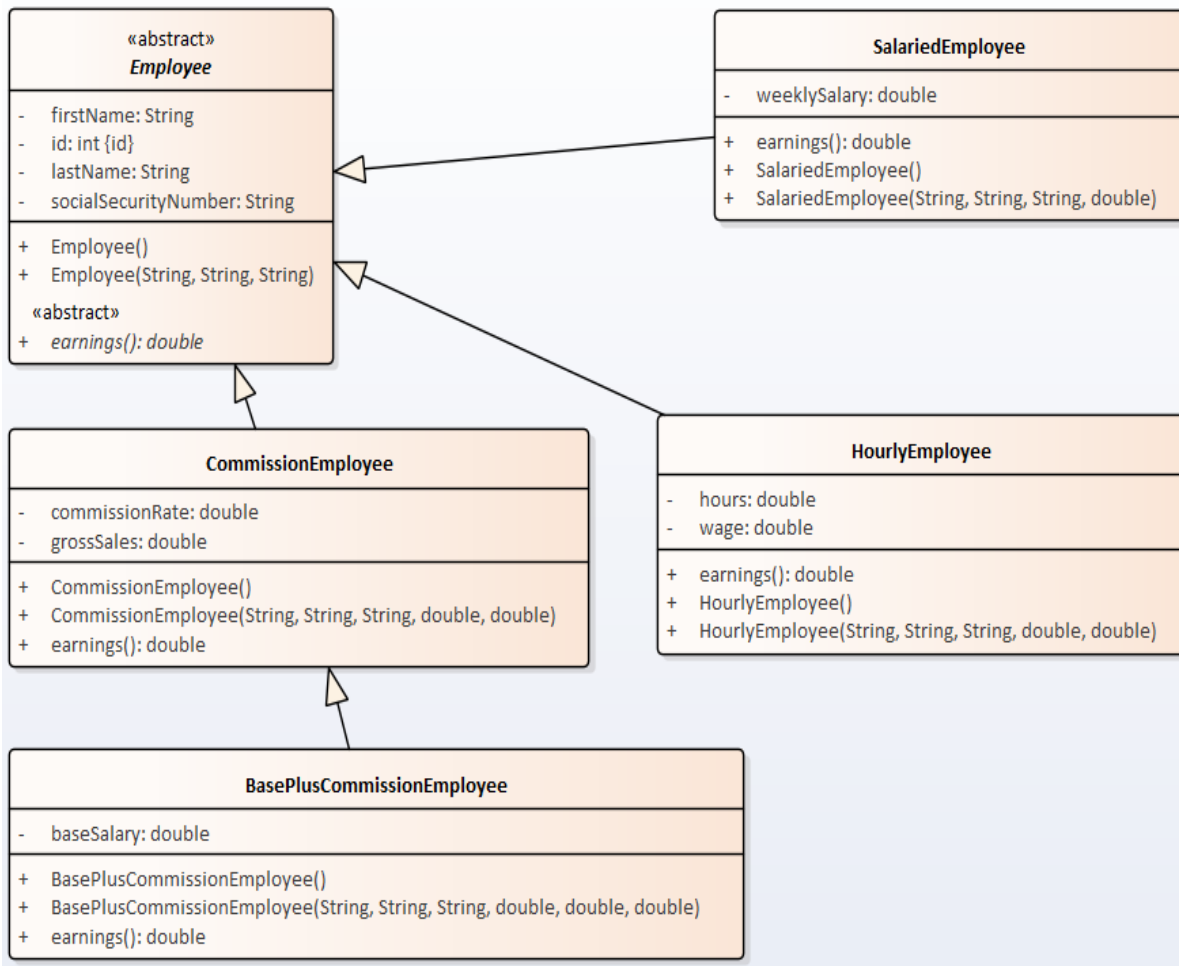
Bài tập 2

Thuộc tính đa trị, mối quan hệ 1 – 1 cùng id, mối quan hệ 1 – n và mối quan hệ n – n

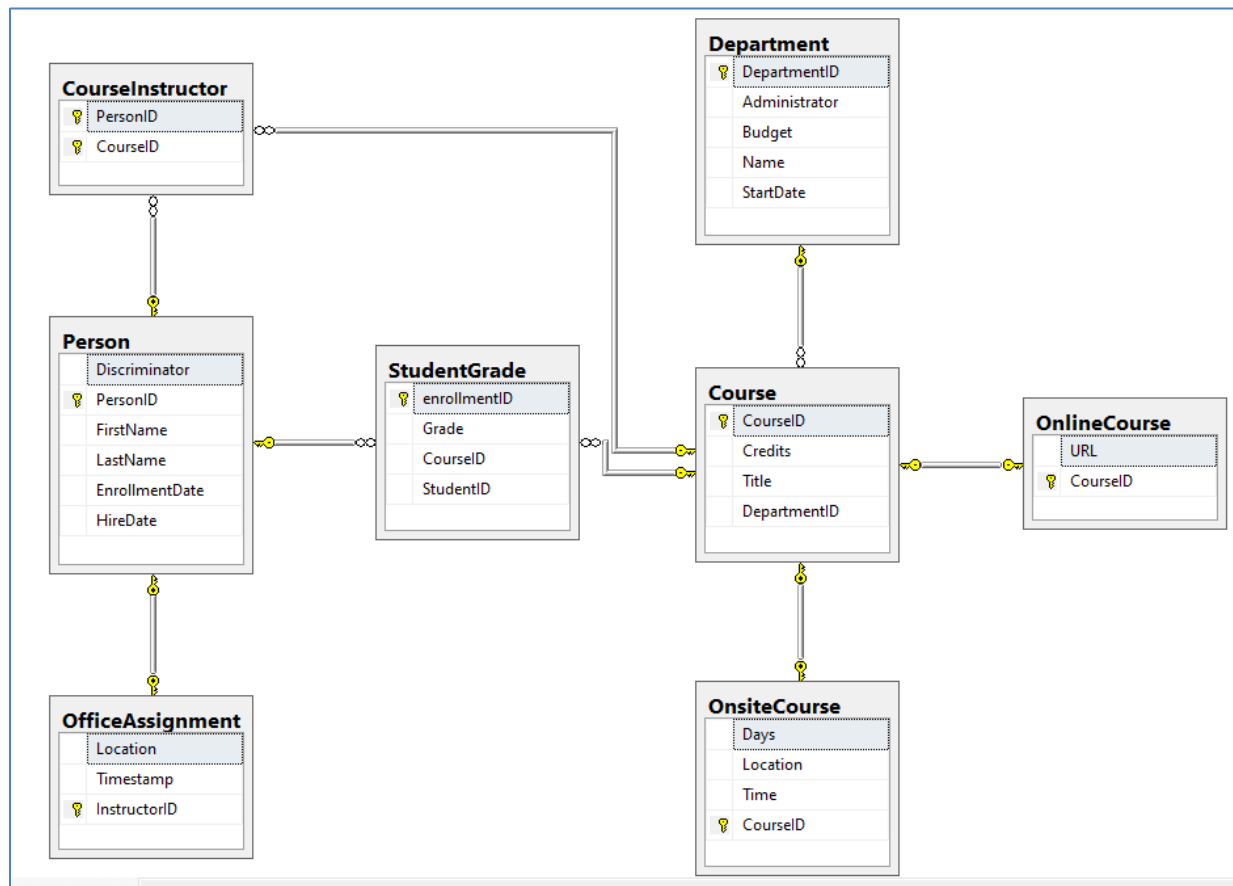
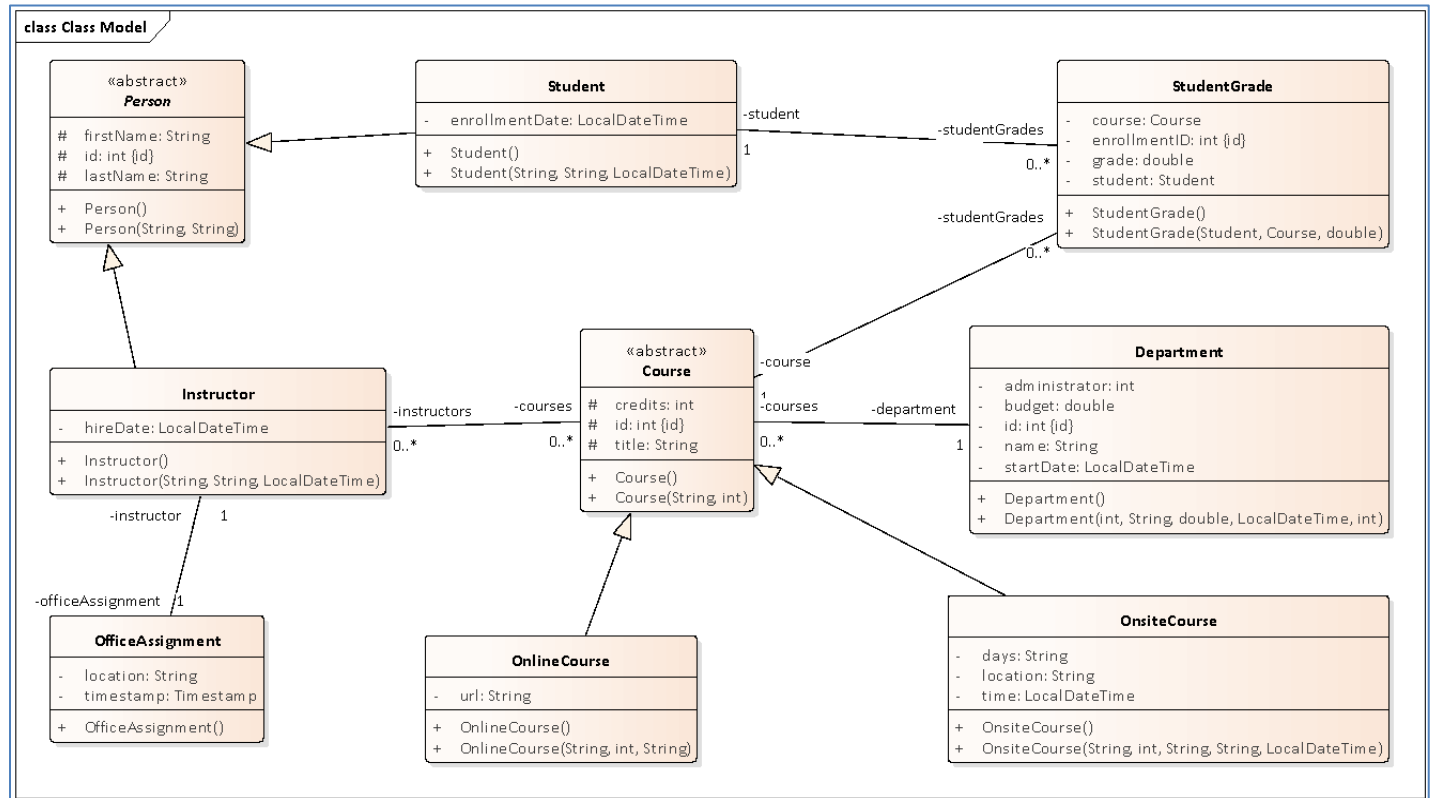


Bài tập 3

Mối quan hệ kế thừa



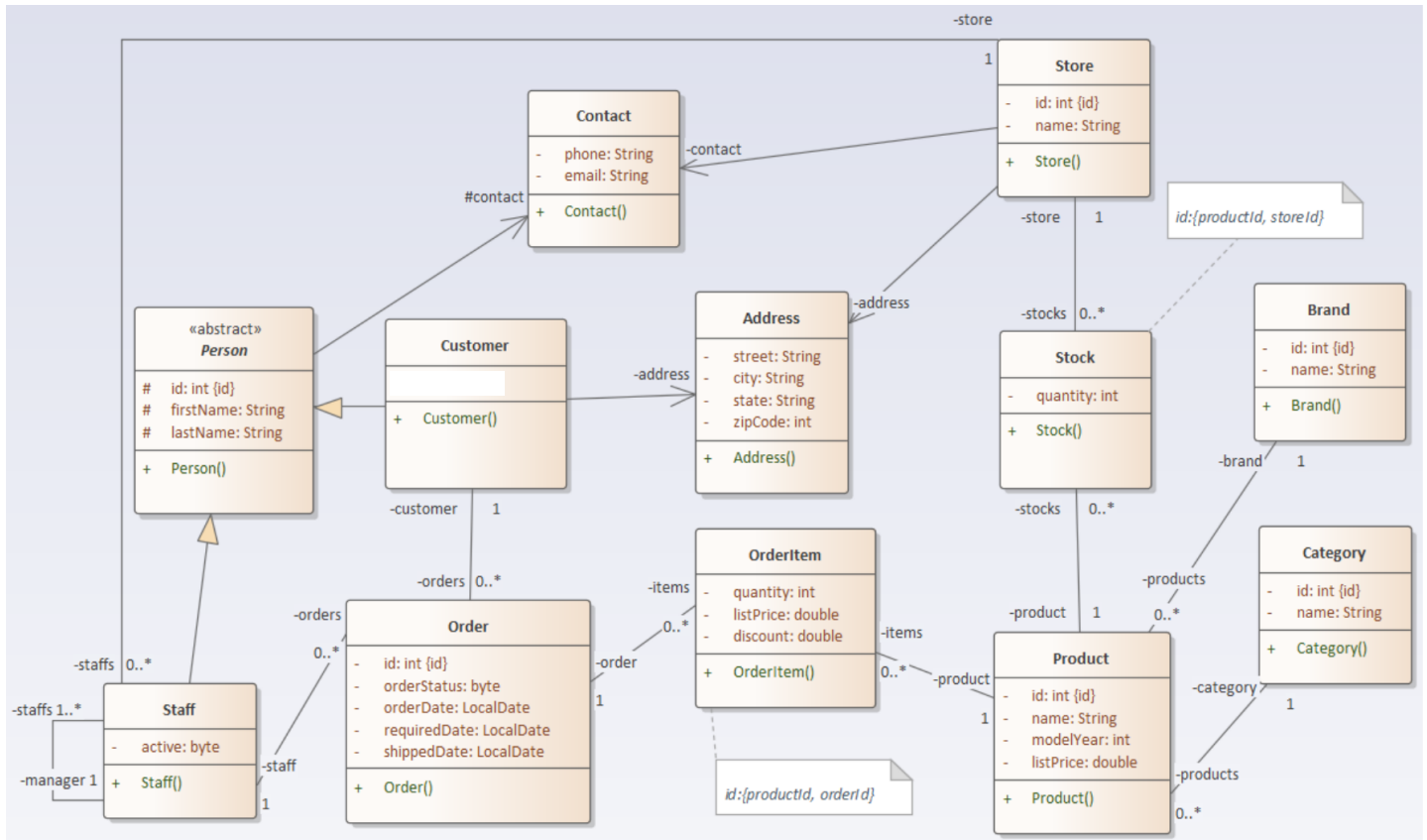
Bài tập 4

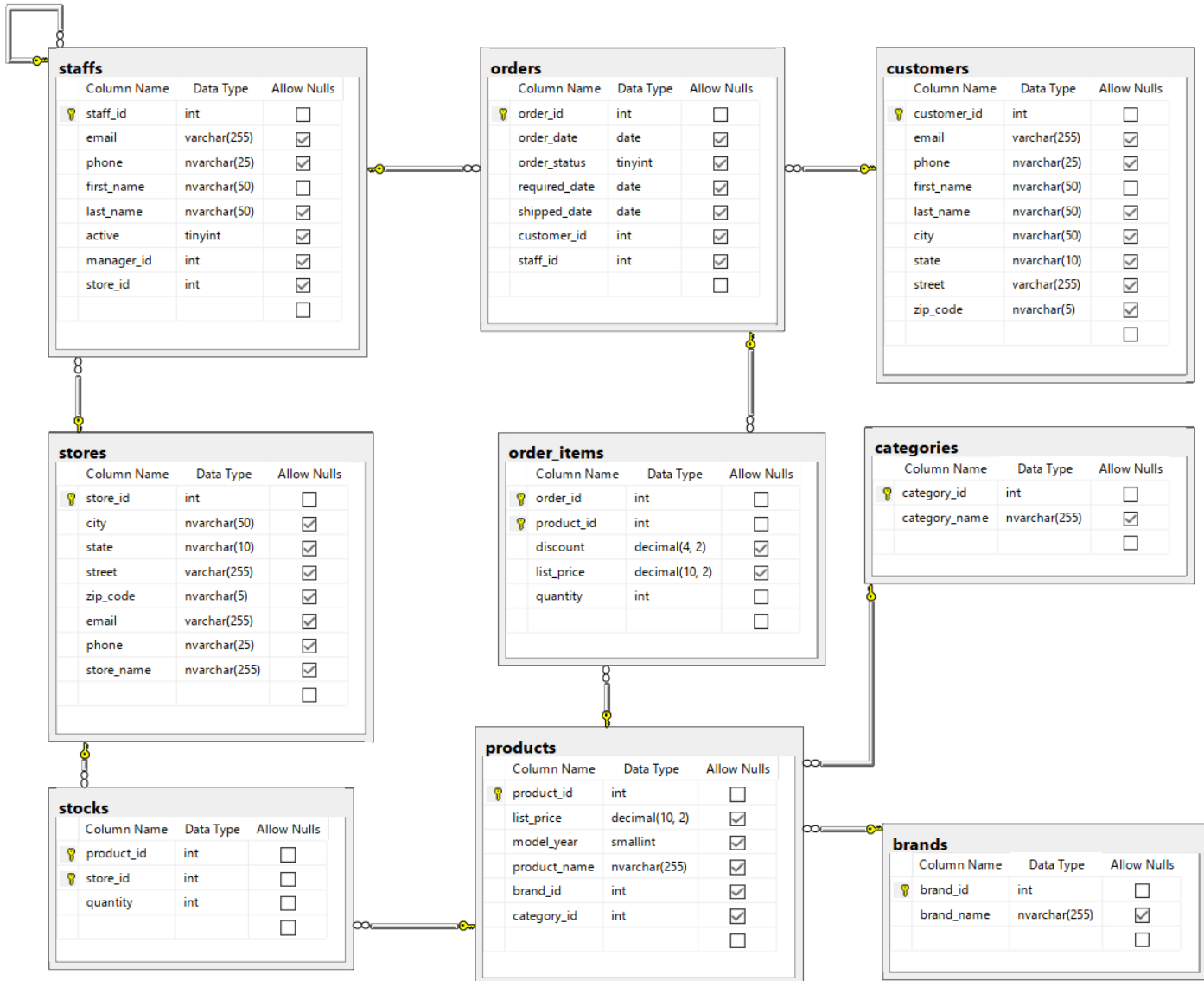


- Ánh xạ từ mô hình lớp sang mô hình CSDL quan hệ trên
- Nhập dữ liệu mẫu
- Thực hiện các thao tác CRUD và tổng hợp dữ liệu bằng ngôn ngữ lập trình Java

Bài tập 5

A) Viết các lớp persistence entity, sau cho ánh xạ ra được mô hình CSDL quan hệ tương ứng:





B) Load dữ liệu database (MS SQL Server)

<https://www.mediafire.com/file/lqjpff4kvrwh4yt/BikeStores.rar>

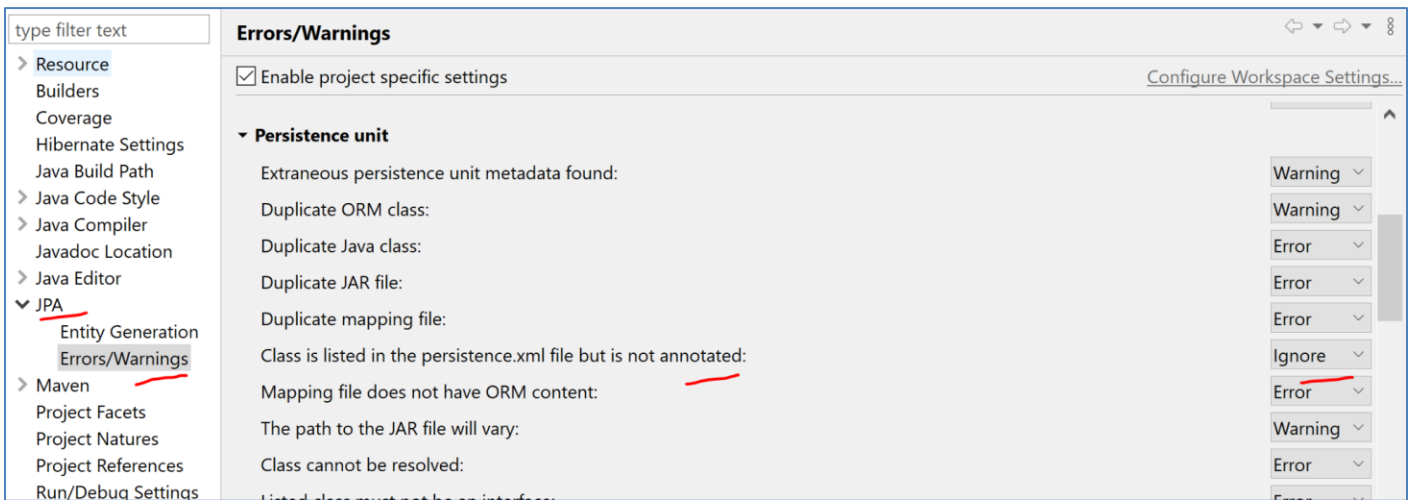
C) (Java Programming) Viết các câu truy vấn sau:

- 1) Thực hiện các thao tác Create, Update, Delete, Find by Id, Get All trên từng đối tượng.
- 2) Tìm danh sách sản phẩm có giá cao nhất.
- 3) Tìm danh sách sản phẩm chưa bán được lần nào.
- 4) Thống kê số khách hàng theo từng bang.
+ getNumberCustomerByState() : Map<String, Integer>
- 5) Tính tổng tiền của đơn hàng khi biết mã số đơn hàng.
- 6) Đếm số đơn hàng của từng khách hàng.
+ getOrdersByCustomers() : Map<Customer, Integer>
- 7) Tính tổng số lượng của từng sản phẩm đã bán ra.
+ getTotalProduct(): Map<Product, Integer>
- 8) Tính tổng tiền của tất cả các hóa đơn trong một ngày nào đó.
- 9) Xóa tất cả các khách hàng chưa mua hàng.
- 10) Thống kê tổng tiền hóa đơn theo tháng / năm.

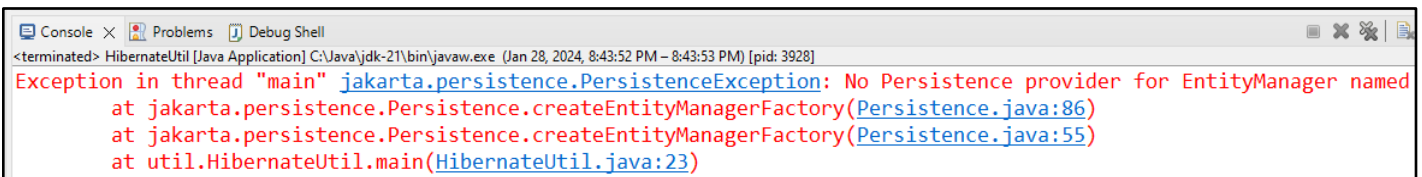
Fix một số các lỗi

Lỗi 1:

Class "entity.Clazz" is listed in the persistence.xml file, but is not annotated

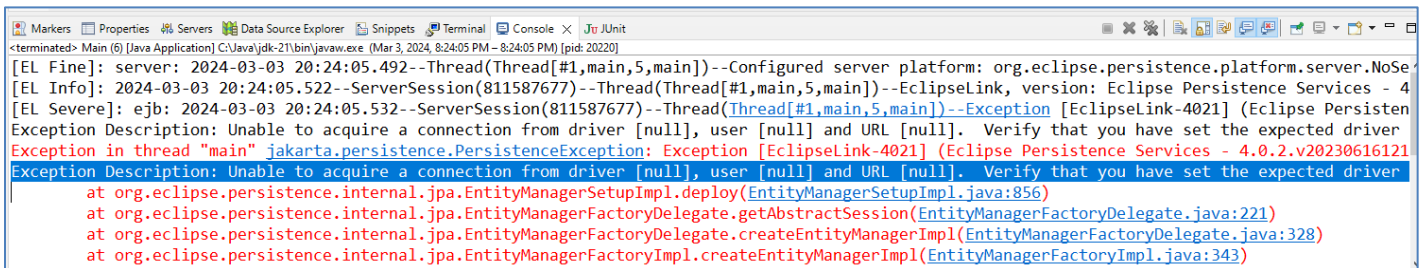


Lỗi 2:



→ Đổi version của hibernate (ví dụ từ 6.4.2 → 6.4.1)

Lỗi 3:



Trong file persistence.xml, thay thế chữ javax thành jakarta

Lỗi 4 MySQL:

(conn=18) Field 'id' doesn't have a default value

→ Run script sau:

SET GLOBAL sql_mode=

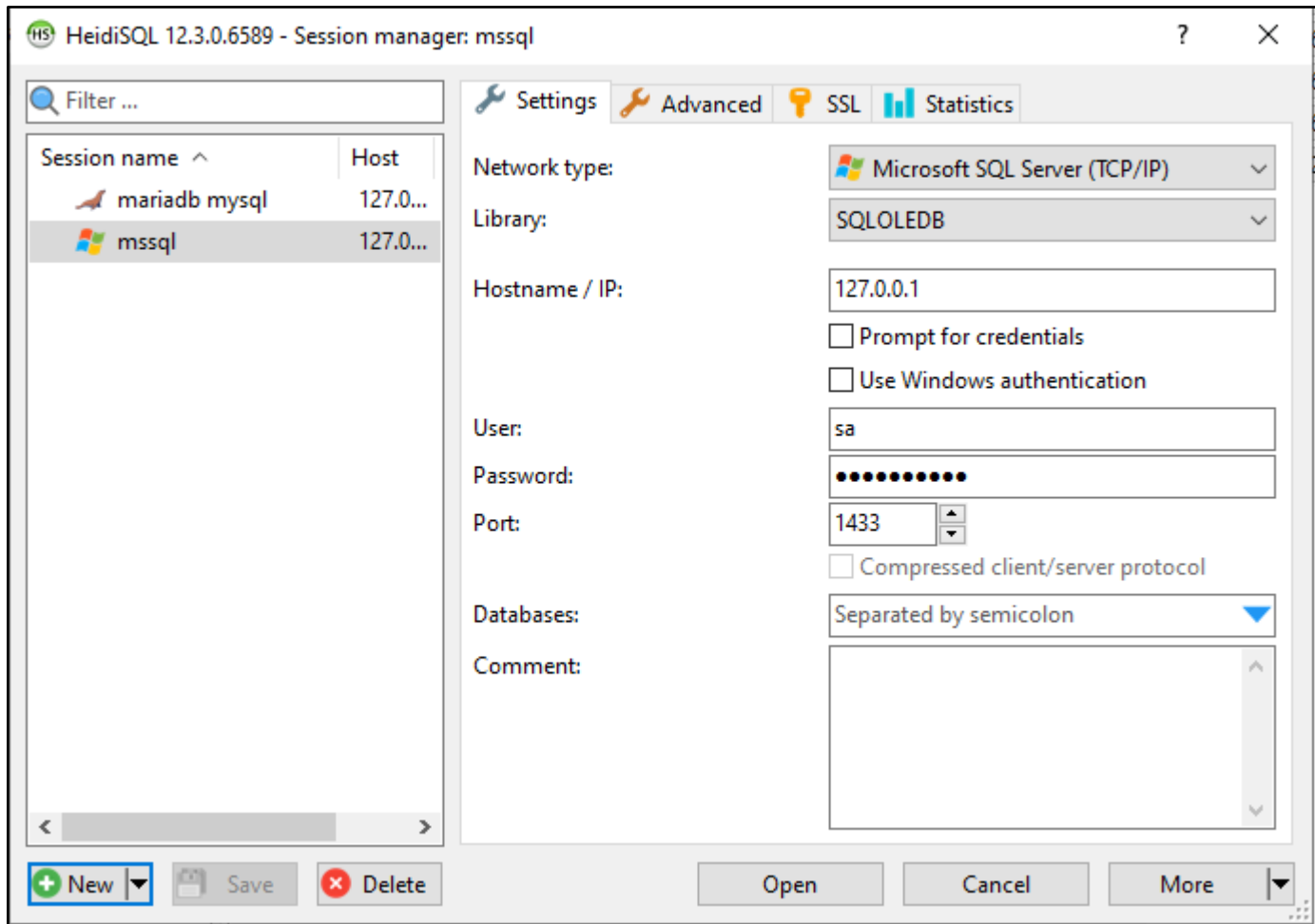
Lỗi 5:

```
Console x
<terminated> Main (9) [Java Application] C:\Java\jdk-21.0.1\bin\javaw.exe (Mar 10, 2024, 10:07:02 PM - 10:07:05 PM) [pid: 7472]
Mar 10, 2024 10:07:04 PM org.hibernate.engine.jdbc.spi.SqlExceptionHelper logExceptions
ERROR: (conn=27) Cannot add or update a child row: a foreign key constraint fails (`testdb`.`enrollments`,
Exception in thread "main" jakarta.persistence.RollbackException: Error while committing the transaction
    at org.hibernate.internal.ExceptionConverterImpl.convertCommitException(ExceptionConverterImpl.java:
    at org.hibernate.engine.transaction.internal.TransactionImpl.commit(TransactionImpl.java:104)
    at test.Main.main(Main.java:31)
Caused by: org.hibernate.exception.ConstraintViolationException: could not execute statement [(conn=27) Ca
    at org.hibernate.exception.internal.SQLExceptionTypeDelegate.convert(SQLExceptionTypeDelegate.java
```

→ Run script sau:

SET GLOBAL FOREIGN_KEY_CHECKS=0;

Connect to MSSQL với HeidiSQL



Cài đặt PostgreSQL

User: postgres Password: postgres

Port: 5432 C:\PostgreSQL

D:\PostgreSQL\data

