

## MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Xây dựng lớp thực thể đơn giản
- ✓ Sử dụng Hibernate/JPA để truy vấn và thao tác dữ liệu cơ bản

## PHẦN I

### BÀI 1 (2 ĐIỂM)

Thực hiện các bước sau đây để chuẩn bị cho dự án làm việc với JPA

- ✓ Tạo CSDL PolyOE và Table Users

```
USE master
GO
CREATE DATABASE PolyOE
GO
USE PolyOE
GO
CREATE TABLE Users(
    Id NVARCHAR(20) NOT NULL,
    Password NVARCHAR(50) NOT NULL,
    Fullname NVARCHAR(50) NOT NULL,
    Email NVARCHAR(50) NOT NULL,
    Admin BIT NOT NULL,
    PRIMARY KEY(Id)
)
```

- ✓ Chèn bằng tay vài bản ghi vào Table Users
- ✓ Tạo dự án Dynamic Web Project
- ✓ Convert sang Maven và khai báo các thư viện phụ thuộc (SQL Server Driver, HibernateJPA)
- ✓ Tạo file persistence.xml trong thư mục [webapp/WEB-INF/classes/META-INF] và cấu hình kết nối đến CSDL PolyOE với thuộc tính persitence unit name là PolyOE

### BÀI 2 (2 ĐIỂM)

- ✓ Tạo lớp thực thể User

```

@Entity
@Table(name = "Users")
public class User {
    @Id
    @Column(name = "id")
    String id;
    @Column(name = "password")
    String password;
    @Column(name = "fullname")
    String fullname;
    @Column(name = "email")
    String email;
    @Column(name = "admin")
    Boolean admin = false;

```

getters/setters

```

}

```

#### ✓ Tạo lớp UserManager

```

public class UserManager {
    EntityManagerFactory factory = Persistence.createEntityManagerFactory("PolyOE");
    EntityManager em = factory.createEntityManager();
    public void findAll() {}
    public void findById() {}
    public void create() {}
    public void update() {}
    public void deleteById() {}
}

```

#### ✓ Viết mã cho các phương thức theo hướng dẫn sau

```

public void findAll() {
    String jpql = "SELECT o FROM User o";
    TypedQuery<User> query = em.createQuery(jpql, User.class);
    List<User> list = query.getResultList();
    list.forEach(user -> {
        String fullname = user.getFullname();
        boolean admin = user.isAdmin();
        System.out.println(fullname + ": " + admin);
    });
}

```

```
public void findById() {
    User user = em.find(User.class, "<<user-id>>");
    String fullname = user.getFullname();
    boolean admin = user.getAdmin();
    System.out.println(fullname + ": " + admin);
}

public void create() {
    User user = new User("U01", "123", "teo@gmail.com", "Tèo", false);
    try {
        em.getTransaction().begin();
        em.persist(user);
        em.getTransaction().commit();
    } catch (Exception e) {
        em.getTransaction().rollback();
    }
}

public void update() {
    User user = em.find(User.class, "U01");
    user.setFullname("Nguyễn Văn Tèo");
    user.setEmail("teonv@gmail.com");
    try {
        em.getTransaction().begin();
        em.merge(user);
        em.getTransaction().commit();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

```
public void deleteById() {  
    User user = em.find(User.class, "U01");  
    try {  
        em.getTransaction().begin();  
        em.remove(user);  
        em.getTransaction().commit();  
    } catch (Exception e) {  
        em.getTransaction().rollback();  
    }  
}
```

- ✓ Tạo lớp UserTest có chứa phương thức main() và chạy thử các phương thức đã xây dựng trong lớp UserManager.

## PHẦN II

### BÀI 3 (2 ĐIỂM)

Hãy viết chương trình console tìm kiếm và xuất họ tên và email của những user có email kết thúc bởi "@fpt.edu.vn" và có vai trò không phải là admin.

*Hướng dẫn:*

- ✓ Viết mã tương tự như `findAll()` nhưng với câu lệnh JPQL là `SELECT o FROM User o WHERE o.email LIKE :search AND o.admin=:role`
- ✓ Sử dụng phương thức `setParameter()` để cung cấp giá trị cho các tham số
  - `query.setParameter("search", "%@fpt.edu.vn");`
  - `query.setParameter("role", false);`

### BÀI 4 (2 ĐIỂM)

Hãy viết chương trình truy vấn và hiển thị danh sách các Users ở trang thứ 3 với yêu cầu kích thước trang (pageSize) là 5 user.

*Hướng dẫn:*

- ✓ Viết mã tương tự như `findAll()`
- ✓ Sử dụng các phương thức `setFirstResult()` để chỉ định vị trí bắt đầu và `setMaxResults()` để chỉ định số phần tử tối đa được truy vấn

- `query.setFirstResult(pageNumber * pageSize); // pageNumber = 2`
- `query.setMaxResults(pageSize);`
- ✓ Chú ý: chèn tối thiểu hơn 15 user vào bảng users trước khi chạy chức năng này

#### **BÀI 5 GIẢNG VIÊN CHO THÊM (2 ĐIỂM)**