MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

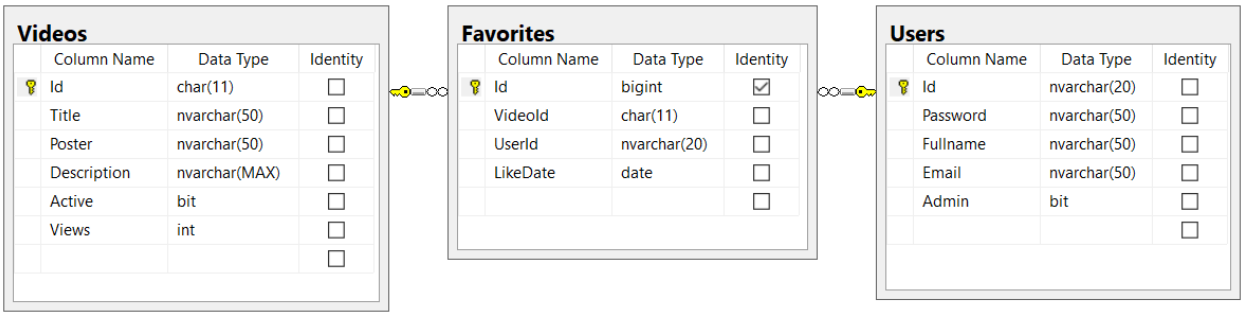
- Ánh xạ các thực thể kết hợp và ràng buộc
- Khai thác được các thực thể kết hợp
- Viết câu lệnh JPQL một cách thành thạo
- Lập trình JPA với SQL và Stored Procedure

PHẦN I: THỰC THỂ KẾT HỢP

BÀI 1: ÁNH XẠ THỰC THỂ KẾT HỢP

Tạo CSDL PolyOE (giải trí trực tuyến) và xây dựng các lớp thực thể mô tả cấu trúc, ràng buộc CSDL.

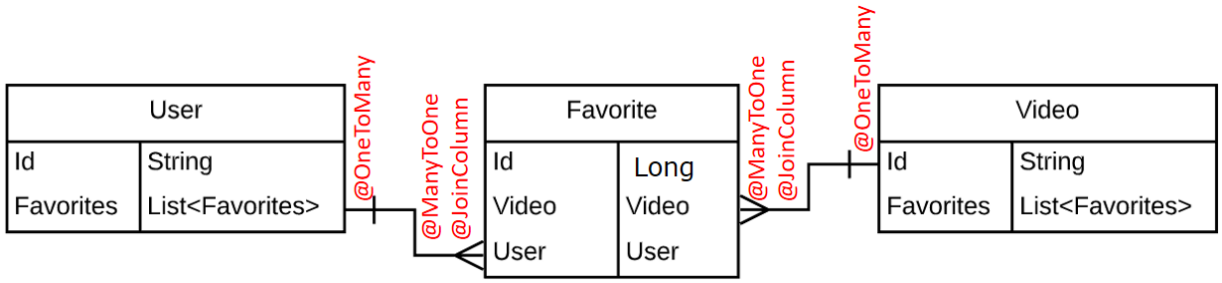
1. Thiết kế CSDL PolyOE như hình sau



Trong đó:

- Videos: Chưa các video giải trí
- Users: chứa người sử dụng
- Favorites: Chứa các videos yêu thích của người sử dụng
- Yêu cầu ràng buộc:
    - Mỗi user chỉ được yêu thích mỗi video một lần (tức là VideoId và UserId phải duy nhất), Favorite Id tự tăng.
    - Không cho phép xóa Video nếu đã được yêu thích
    - Không cho phép xóa user nếu đã có yêu thích

2. Xây dựng các thực thể mô tả CSDL và các ràng buộc của nó.



Trong đó: Favorite (Video và User) là duy nhất, Favorite Id tự tăng.

- Lớp thực thể User

```
@Entity
@Table(name = "Users")
public class User {
        @Id
        String id;
        String password;
        String fullname;
        String email;
        Boolean admin = false;
        @OneToMany(mappedBy = "user")
        List<Favorite> favorites;
        getters/setters
}
```

- Lớp thực thể Video

```
@Entity
@Table(name = "Videos")
public class Video {
        @Id
        String id;
        String title;
        String poster;
        String description;
        Integer views = 0;
        Boolean active = true;
        @OneToMany(mappedBy = "video")
        List<Favorite> favorites;
        getters/setters
}
```

- Lớp thực thể Favorite

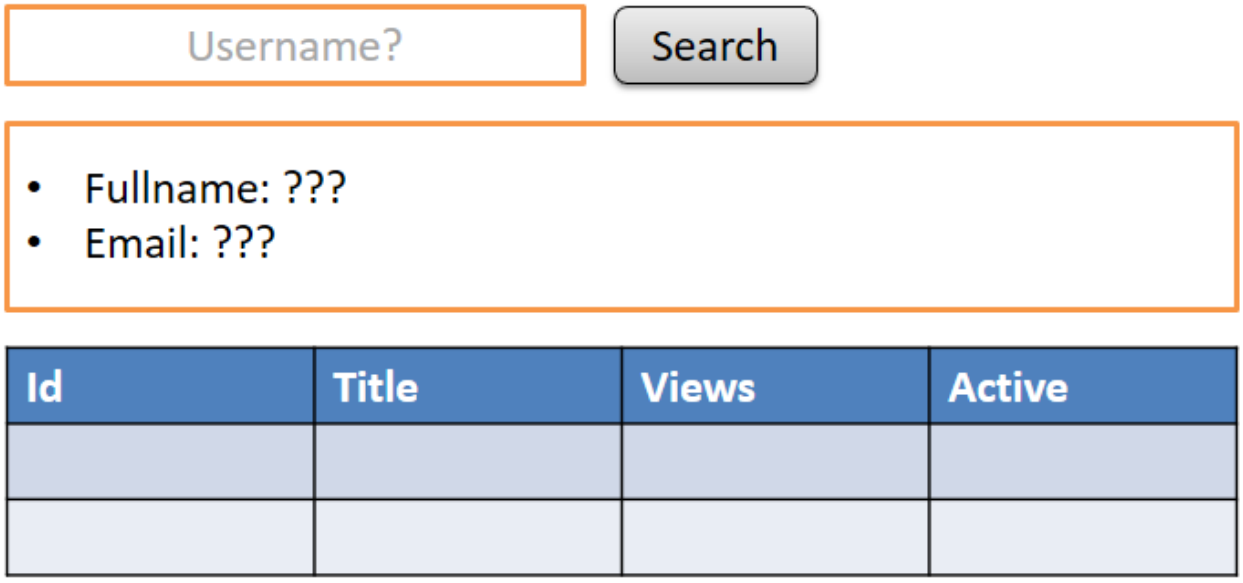
```
@Entity
```

```
@Table(name = "Favorites", uniqueConstraints={
                @UniqueConstraint(columnNames = {"VideoId" , "UserId"})
})
public class Favorite {
        @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
        Long id;
        @ManyToOne  @JoinColumn(name = "UserId")
        User user;
        @ManyToOne  @JoinColumn(name = "VideoId")
        Video video;
        @Temporal(TemporalType.DATE)
        Date likeDate = new Date();
        getters/setters
}
```

BÀI 2: LẬP TRÌNH KHAC THÁC THỰC THỂ KẾT HỢP

Hãy xây dựng các trang web để khai thác quan hệ thực thể kết hợp

1. Tìm các video yêu thích theo người sử dụng (nhập user id)



Hướng dẫn:

```
String username = req.getParameter("username");
User user = em.find(User.class, username);
List<Favorite> favorites = user.getFavories();

req.setAttribute("user", user);
req.setAttribute("favorites", favorites);
```

2. Tìm các video được yêu thích có title chứa từ khóa (nhập từ khóa)

Id	Title	Views	Active

Hướng dẫn:

```
String keyword = req.getParameter("keyword");

String jpql = "SELECT DISTINCT o.video FROM Favorite o "
            + " WHERE o.video.title LIKE :keyword";
TypedQuery<Video> query = em.createQuery(jpql, Video.class);
query.setParameter("keyword", keyword);
List<Video> list = query.getResultList();

req.setAttribute("videos", list)
```

3. Tìm những người sử dụng thích video (nhập video id)

Username	Fullname	Email	Role

Hướng dẫn:

```
String videoId = req.getParameter("videoId");

String jpql = "SELECT o.user FROM Favorite o WHERE o.video.id=:vid";
TypedQuery<User> query = em.createQuery(jpql, User.class);
query.setParameter("vid", videoId);
List<User> list = query.getResultList();
```

```
req.setAttribute("users", list)
```

4. Hiển thị tất cả các video không có hoặc có yêu thích

Id	Title	Views	Active?

```
boolean favorite = Boolean.parseBoolean(req.getParameter("favorite"));

String jpql = "SELECT o FROM Video o WHERE o.favorites IS EMPTY";
if(favorite) {
        jpql = "SELECT o FROM Video o WHERE o.favorites IS NOT EMPTY";
}
TypedQuery<Video> query = em.createQuery(jpql, Video.class);
List<Video> list = query.getResultList();

req.setAttribute("videos", list)
```

5. Tổng hợp số lượt thích từng video theo cấu trúc (video title, số lượt thích, ngày thích lâu nhất, ngày thích mới nhất)

Id	Favorite Count	Newest Date	Oldest Date

Hướng dẫn:

- Xây dựng thêm một lớp thực thể chứa kết quả truy vấn thông tin tổng hợp

```
@Entity
public class Report {
	@Id
	Serializable group;
	Long likes;
	Date newest;
	Date oldest;
	public Report(Serializable group, Long likes, Date newest, Date oldest) {
		this.group = group;
		this.likes = likes;
		this.newest = newest;
		this.oldest = oldest;
	}
	getters/setters
}
```

- Sử dụng lớp thực thể Report để nắm giữ thông tin tổng hợp

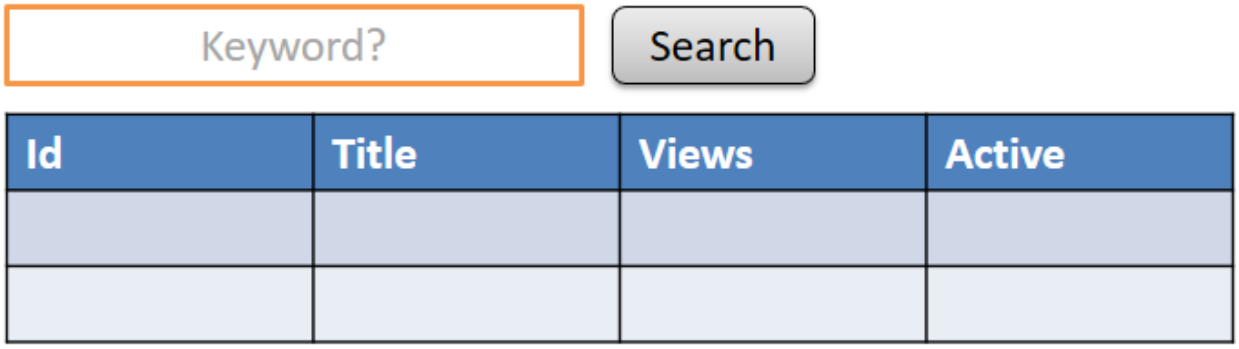
```
String jpql = "SELECT new Report(o.video.title, count(o), "
		+ " max(o.likeDate), min(o.likeDate)) "
		+ " FROM Favorite o "
		+ " GROUP BY o.video.title ";
TypedQuery<Report> query = em.createQuery(jpql, Report.class);
List<Report> list = query.getResultList();
```

PHẦN II: LẬP TRÌNH JPA NÂNG CAO

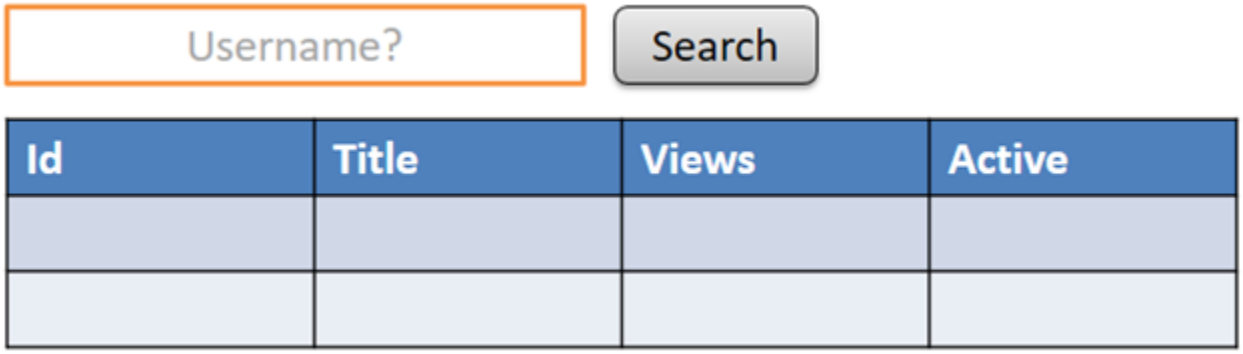
BÀI 3: TRUY VẤN THỰC THỂ NÂNG CAO

Hãy lập trình JPA với @NamedQuery để thực hiện các yêu cầu truy vấn sau đây

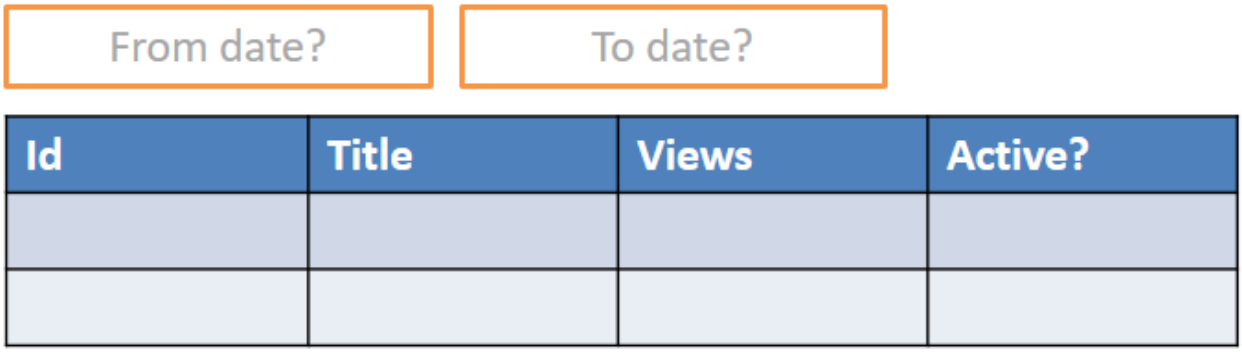
2. Tìm các video được yêu thích có title chứa từ khóa (nhập từ khóa)



2. Tìm các video yêu thích theo người sử dụng (nhập user id)



3. Tìm những video được thích trong khoảng thời gian (nhập min date, max date)



4. Tìm những video được thích trong trong các tháng (nhập danh sách tháng)

☒ 1 ☐ 2 ☒ 3 ☒ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10 ☒ 11 ☒ 12

Id	Title	Views	Active?

Hướng dẫn

- Khai báo các câu lệnh truy vấn trên các thực thể truy vấn và đặt tên cho các câu lệnh truy vấn đó

```
@NamedQueries({
        @NamedQuery(name = "Video.findByKeyword",
                query = "SELECT DISTINCT o.video FROM Favorite o"
                        + " WHERE o.video.title LIKE :keyword"),
        @NamedQuery(name = "Video.findByUser",
                query = "SELECT o.video FROM Favorite o"
                        + " WHERE o.user.id=:id"),
        @NamedQuery(name = "Video.findInRange",
                query = "SELECT DISTINCT o.video FROM Video o"
                        + " WHERE o.likeDate BETWEEN :min AND :max"),
        @NamedQuery(name = "Video.findInMonths",
                query = "SELECT DISTINCT o.video FROM Video o"
                        + " WHERE month(o.likeDate) IN (:months)")
})
@Entity
@Table(name = "Videos")
public class Video {...}
```

- Sử dụng các JPQL đã đặt tên như sau

```
String keyword = req.getParameter("keyword");

TypedQuery<Video>query=em.createNamedQuery("Video.findByKeyword",Video.class);
query.setParameter("keyword", keyword);
List<Video> list = query.getResultList();

req.setAttribute("videos", list)
```

- Đối với toán tử IN () thì tham số của nó phải là List<T>. Như vậy muốn sử dụng **Video.findInMonths** thì tham số của nó phải là **List<Integer>**

```
String[] values = req.getParameterValues("month");
List<Integer> months = new ArrayList<Integer>();
```

```
for(String month: values) {
        list.add(Integer.valueOf(month));
}
TypedQuery<Video>query=em.createNamedQuery("Video.findInMonths",Video.class);
query.setParameter("months", months);
List<Video> list = query.getResultList();

req.setAttribute("videos", list)
```

BÀI 4: SỬ DỤNG SQL VÀ STORED PROCEDURE

JPA cho phép lập trình với SQL và Stored Procedure để tận dụng những điểm mạnh của hệ quản trị CSDL đặt thù. Trong phần này yêu cầu thực hiện 2 hành động sau

1. Truy vấn và hiển thị 10 video ngẫu nhiên (sử dụng SQL)

Id	Title	Views	Active?

Hướng dẫn:

- Thêm annotation vào trên lớp thực thể Video

```
@NamedNativeQueries({
        @NamedNativeQuery(name = "Report.random10",
        query = "SELECT TOP 10 * FROM Videos ORDER BY newid()",
        resultClass = Video.class)
})
```

- Truy vấn sử dụng câu lệnh đã đặt tên

```
Query query = em.createNamedQuery("Report.random10");
List<Video> list = query.getResultList();
```

2. Tổng hợp số lượt thích từng video theo năm có cấu trúc (video title, số lượt thích, ngày thích lâu nhất, ngày thích mới nhất) (Sử dụng Stored Procedure)

Title	Favorite Count	Newest Date	Oldest Date

Hướng dẫn:

- Tạo Stored Procedure như sau

```
CREATE PROC spFavoriteByYear(@Year INT)
AS
BEGIN
	SELECT
		v.Title AS 'Group',
		count(f.Id) AS 'Likes',
		max(f.LikeDate) AS 'Newest',
		min(f.LikeDate) AS 'Oldest'
	FROM Favorites f JOIN Videos v ON v.Id = f.VideoId
	WHERE year(f.LikeDate) = @Year
	GROUP BY v.Title
END
```

- Khai báo annotation vào lớp thực thể Report

```
@NamedStoredProcedureQueries({
	@NamedStoredProcedureQuery(name = "Report.favoriteByYear",
			procedureName = "spFavoriteByYear",
			parameters = {
			@StoredProcedureParameter(name="year", type = Integer.class)},
			resultClasses = {Report.class})
})
@Entity
public class Report {...}
```

- Lập trình JPA gọi Stored Procedure

```
Integer year = Integer .valueOf(req.getParameter("year"));

StoredProcedureQuery query =
			em.createStoredProcedureQuery("Report.favoriteByYear");
query.setParameter("year", year);
List<Video> list = query.getResultList();
```