



## **JAVA PERSISTENCE QUERY LANGUAGE**

---

### **LẬP TRÌNH JAVA #4 (P4.2)**

- @NamedQuery
- @NativeQuery
- @NamedNativeQuery
- @NamedStoredProcedureQuery



- ❑ JPA cho phép khai báo trước các câu lệnh JPQL gắn liền với từng thực thể để tiện cho việc quản lý và nâng cấp.
- ❑ Mỗi câu lệnh JPQL phải được đặt tên, khai báo tham số và chỉ ra kết quả trả về

```
@NamedQueries({  
    @NamedQuery(name = "User.findAll", query = "SELECT o FROM User o"),  
    @NamedQuery(name = "User.findByEmail",  
        query = "SELECT o FROM User o WHERE o.email LIKE :email"),  
    @NamedQuery(name = "User.findByRole",  
        query = "SELECT o FROM User o WHERE o.admin=:role"),  
})  
@Entity  
@Table(name = "Users")  
public class User {...}
```

```
TypedQuery<User> query = em.createNamedQuery("User.findAll", User.class);  
List<User> list = query.getResultList();
```

```
TypedQuery<User> query = em.createNamedQuery("User.findByRole", User.class);  
query.setParameter("role", true);  
List<User> list = query.getResultList();
```

```
TypedQuery<User> query = em.createNamedQuery("User.findByEmail", User.class);  
query.setParameter("email", "nghiemn@fpt.edu.vn");  
User entity = query.getSingleResult();
```

- Gọi **createNamedQuery(name)** và truyền tên của query đã đặt tên trong Entity để tạo Query thay vì gọi *createQuery(jpql)* như thông thường



# DEMOSTATION

---

❑ SQL đặc thù là câu lệnh SQL phụ thuộc vào hệ quản trị CSDL. JPA cho phép truy vấn dữ liệu với câu lệnh SQL đặc thù.

❑ Ưu điểm:

- ❖ Tận dụng kiến thức SQL sẵn có của lập trình viên, không cần phải học thêm JPQL.
- ❖ Tận dụng được sức mạnh của hệ quản trị của CSDL

❑ Hạn chế:

- ❖ Thay đổi hệ quản trị CSDL thì phải viết lại code của dự án làm cản trở việc nâng cấp, mở rộng phần mềm về sau.

```
String sql = "SELECT * FROM Users WHERE email LIKE ?";  
Query query = em.createNativeQuery(sql, User.class);  
query.setParameter(1, "%@gmail.com");  
List<User> list = query.getResultList();
```

- ❑ Sử dụng SQL thay vì JPQL.
- ❑ Gọi createNativeQuery(sql) thay vì createQuery(jpql) như trước đây.

```
@NamedNativeQueries({  
    @NamedNativeQuery(  
        name = "Users .findByEmail",  
        query = "SELECT * FROM Users WHERE email LIKE ?"  
        resultClass = User.class  
    )  
})  
@Entity  
@Table(name = "Users")  
public class User {...}
```

```
Query query = em.createNamedQuery("Users.findByEmail");  
query.setParameter(1, "%@gmail.com");  
List<User> list = query.getResultList();
```



# DEMOSTATION

---

- ❑ JPA cho phép gọi PROC với mục đích
  - ❖ Tận dụng hệ thống PROC sẵn có của CSDL
  - ❖ Tận dụng sức mạnh của hệ quản trị CSDL
  - ❖ Đơn giản hóa code trong lập trình
- ❑ Tương tự với NativeQuery, PROC phụ thuộc vào hệ quản trị CSDL nên khi thay đổi hệ quản trị CSDL thì phải viết lại PROC.

- ❑ Ví dụ: PROC sp\_FindByEmail

```
CREATE PROCEDURE sp_FindByEmail(@Email NVARCHAR(50))  
AS  
BEGIN  
    SELECT * FROM Users WHERE email LIKE @Email  
END
```

```
@NamedStoredProcedureQueries({  
    @NamedStoredProcedureQuery(  
        name = "User.spFindByEmail",  
        procedureName = "sp_FindByEmail",  
        resultClasses = {User.class},  
        parameters = @StoredProcedureParameter(name = "email", type = String.class)  
    )  
})  
@Entity  
@Table(name = "Users")  
public class User {...}
```

```
StoredProcedureQuery query =  
    em.createNamedStoredProcedureQuery("User.spFindByEmail");  
query.setParameter("email", "%@gmail.com");  
List<User> list = query.getResultList();
```



# DEMOSTATION

---

- ✓ @NamedQuery
- ✓ @NativeQuery
- ✓ @NamedNativeQuery
- ✓ @NamedStoredProcedureQuery





Cảm ơn