

Creating Database Queries With Named Queries

By Ramesh Fadatare (Java Guides)

Named Queries

- Named queries are one of the core concepts in JPA. They enable you to declare a query in your persistence layer and reference it in your business code. That makes it easy to reuse an existing query.

```
@Entity
@NamedQuery(name = "Product.findBySku",
            query = "SELECT p from Product p WHERE p.sku =:sku"
)
public class Product {
}
```

- When you define a named query, you can provide a JPQL query or a native SQL query in very similar ways.
- If we want to create a JPQL query, we have to annotate our entity with the `@NamedQuery` annotation. If we want to create a SQL query, we have to annotate our entity with the `@NamedNativeQuery` annotation.

Steps to Define Named JPQL Query

If we want to create a JPQL query, we must follow these steps:

- Annotate the entity with the `@NamedQuery` annotation from JPA.
- Use `@NamedQuery` annotation's **name** attribute to set name of the named query (`Product.findBySku`)
- Use `@NamedQuery` annotation's **query** attribute to set the JPQL query (`SELECT p from Product p WHERE p.sku =:sku`) as the value

```
@Entity
@NamedQuery(name = "Product.findBySku",
            query = "SELECT p from Product p WHERE p.sku =:sku"
)
public class Product {
}
```

- Use named query name in a Repository

```
// named query method
Product findBySku(@Param("sku") String sku);
```


Define Multiple Named JPQL Queries

If we want to create a multiple JPQL query, we must follow these steps:

- Annotate the entity with the `@NamedQueries` annotation from JPA/Hibernate.
- Use multiple `@NamedQuery` annotations from JPA/Hibernate to define a named queries
- Set the name of the named query as the value of the `@NamedQuery` annotation's **name** attribute.
- Set the JPQL query as the value of the `@NamedQuery` annotation's **query** attribute.

```
@NamedQueries(  
    {  
        @NamedQuery(name = "Product.findAllOrderByNameDesc",  
                    query = "SELECT p from Product p ORDER BY p.name DESC"  
        ),  
        @NamedQuery(name = "Product.findByName",  
                    query = "SELECT p from Product p WHERE p.name =:name"  
        )  
    }  
)
```


Steps to Define Named SQL Query

If we want to create a SQL query, we must follow these steps:

- Annotate the entity with the `@NamedNativeQuery` annotation from JPA.
- Set the name of the named query as the value of the `@NamedNativeQuery` annotation's **name** attribute.
- Set the SQL query as the value of the `@NamedNativeQuery` annotation's **query** attribute.
- Set the returned entity class (`Product.class`) as the value of the `@NamedNativeQuery` annotation's **resultClass** attribute.

```
@NamedNativeQuery(  
    name = "Product.findBySku",  
    query = "SELECT * from PRODUCTS WHERE sku =:sku",  
    resultClass = Product.class  
)
```

Using named native query in Repository

```
// named native query method  
@Query(nativeQuery = true)  
Product findBySku(@Param("sku") String sku);
```


Define Multiple Named SQL Queries

If we want to create a multipleJPQL query, we must follow these steps:

- Annotate the entity with the `@NamedNativeQueries` annotation from JPA/Hibernate.
- Use multiple `@NamedNativeQuery` annotations from JPA/Hibernate to define a named native queries
- Set the values to name, query and resultClass attributes of `@NamedNativeQuery` annotation