

## 8.2.2. JPQL - @Query

上一章節介紹了 Query 物件，在這章再延伸討論 @Query，@Query 可以做到和 Query 類似的事情，但寫法更加簡潔，不過也有一些限制，例如比較適用於單表查詢，如果是有 join 的情況，比較適合使用下一章的 **8.3. JPQL - Query with SQL File**。

和上一章提到的一樣，如果想要在 SQL 加入使用者傳進來的參數的話，可以在 SQL 語句傳入動態查詢參數，有以下幾種方法：

1. 依位置傳入參數 ( Positional Parameters )：類似 PreparedStatement 用法

```
"select * from STUDENT.TB_EMP where EMP_ID = ?1"
```

2. 依參數名稱對照 ( Named Parameters )

```
"select * from STUDENT.TB_EMP where EMP_ID = :EMP_ID"
```

接下來的練習我們會使用第二種方式，也就是依照參數名稱將對應的參數傳入查詢條件。

```
|--com.example.demospringboot
|   |--DemospringbootApplication.java
|--com.example.demospringboot.configuration
|   |--SwaggerConfig.java
|   |--RestConfiguration.java
|--com.example.demospringboot.controller
|   |--TestController.java // 修改的檔案
|   |--ProductController.java
|   |--EmpController.java
|--com.example.demospringboot.entity
|   |--Car.java
|   |--CarPK.java
|   |--Emp.java
|--com.example.demospringboot.model
|   |--Product.java
|--com.example.demospringboot.repository
|   |--CarRepository.java // 修改的檔案
|--com.example.demospringboot.service
|   |--CarService.java // 修改的檔案
|   |--ProductService.java
|--com.example.demospringboot.service.impl
|   |--CarServiceImpl.java // 修改的檔案
|   |--ProductServiceImpl.java
|--com.example.demospringboot.service.sql
|   |--SqlAction.java
```

先修改 `CarRepository.java` 的內容：在這支檔案新增 `existsByManufacturerAndPriceMore(...)` 方法，傳入三個參數 `manufacturer` 、 `minPrice` 、 `price` ，每個參數前，需要多加 `@Param(...)` ，括號內的參數，會對應再後面 `@Query` 的變數。

然後再方法上放上 `@Query` ，`@Query` 後面放進 SQL 語法，會動態更動的值同樣的用 **:變數名** 取代，這樣 JPA 才能把參數值替換到 SQL 內。

```
@Query(" select case when count(*)> 0 then 'true' else 'false' end as exist from CarEntity C "
      + "where C.manufacturer = :manufacturer and (C.minPrice > :minPrice or C.price > :price)")
boolean existsByManufacturerAndPriceMore(
    @Param("manufacturer") String manufacturer,
    @Param("minPrice") BigDecimal minPrice,
    @Param("price") BigDecimal price);
```

SQL 撰寫需要注意，因為該 table 已經交由 `@Entity` 管理，在使用上我們無法直接去跟資料庫溝通，而是去找持久層，看上方的 SQL 語法可以看到，原本的 table 名稱是使用 **CarEntity**，而後面的欄位同樣地是指 **CarEntity** 內的屬性。

如果在 `CarEntity` 內設定 `@Entity(name = "XXXXXX")`，那麼 SQL 語法應該應該也要改成：

```
@Query(" select case when count(*)> 0 then 'true' else 'false' end as exist from XXXXXX C "
      + "where C.manufacturer = :manufacturer and (C.minPrice > :minPrice or C.price > :price)")
```

想要繼續使用原生 SQL，只需要在 `@Query` 註解中設定 `nativeQuery=true`，然後 SQL 調整成 原生 SQL 即可。

最後再來修改 `Controller` 及 `Service` 的部分，呼叫 `exist(...)` 因為只有要看效果，值不從 request 拿，直接寫在程式碼內。

- `CarController.java`

```
@RequestMapping(value = "/exist", method = RequestMethod.GET)
public boolean exist() throws ErrorInputException, DataNotFoundException {
    return carService.exist("adadad", new BigDecimal(10), new BigDecimal(100));
}
```

- `CarService.java`

```
boolean exist(String manufacturer, BigDecimal minPrice, BigDecimal price);
```

- `CarServiceImpl.java`

```
@Override
public boolean exist(String manufacturer, BigDecimal minPrice, BigDecimal price) {
    return carRepository.existsByManufacturerAndPriceMore(manufacturer, minPrice, price);
}
```

修改完後打 url 測試就可以拿到相對應的資料。

## 參考

<https://openhome.cc/Gossip/EJB3Gossip/Query.html>

<https://www.baeldung.com/jpa-query-parameters>