Java WebApi Coding Exercise

Car Collectors is a collection of API for users to gather information about cars and those owned by users of a Car selling site.

The scope of the exercise is as follows:

1. Create a database for CarCollection in MySql (or any other relational database you prefer). Create tables for `Car`, `User` and `Manufacturer` that is connected to each other via foreign keys and serve the purpose of the below mentioned APIs. Try to write optimized SQL queries.

2. Provide an endpoint for a `Cars` resource that returns all `Car` models.

- Optional query parameters:
  - `year (int)` *(cars built in this specific year)*
  - `priceFrom (double)` *(cars ranging from this price)*
  - `priceTo (double)` *(cars ranging to this price)*
  - `manufacturerId (int)` *(cars belonging to this manufacturer)*
- Sample output:


```
[
  {
    "id": "12",
    "manufacturerName": "Audi",
    "model": "RS3",
    "bodyType": "Saloon",
    "year": 2019,
    "retailPrice": 46500.99
  }
]
```


2.a Provide an endpoint for the `Cars` resource that returns a single `Car` model when given the `Id` of the `Car` in the `Route` of the request URI.

The sample output should be the same as in point 2 but instead of a collection be a single object.

2.b Provide an endpoint for the `Cars` resource that creates a new `Car`.

The sample output should be the same as both 2 and 2a. You should decide the input for the request and ensure that it is able to create a car that is associated with a manufacturer. Also try to put some validations may be required when creating a new car**.

3. Provide an endpoint for a `Users` resource that returns all `User` models along with the `Car` entities that they own.

- Optional query parameters:

    - `hasCarCount (int)` *(users that have X number of cars)*

- Sample output:

```
[
  {
    "id": "11",
    "fullName": "Ulrich Nielsen",
    "postCode": "SW1P 4QE",
    "cars": [
      {
        "id": "12",
        "plateNumber": "RS03 ABC",
        "manufacturerName": "Audi",
        "model": "RS3",
        "bodyType": "Saloon",
        "year": 2019,
        "purchasePrice": 41250.00,
        "retailPrice": 46509.00
      }
    ]
  }
]
```


4. Provide an endpoint for a `Manufacturers` resource that returns all `Manufacturer` models along with the `Car` entities they produce.

- Optional query parameters:
    - `name (string)` *(name of the manufacturer)*
    - `year (int)` *(cars built in this specific year)*

- Sample output:

```
[
  {
    "id": "4",
    "name": "Audi",
    "country": "Germany",
    "cars": [
      {
        "id": "2",
        "manufacturerName": "Audi",
        "model": "Q7",
        "bodyType": "SUV",
        "year": 2016,
        "retailPrice": 52670.00
      }
    ]
  }
]
```