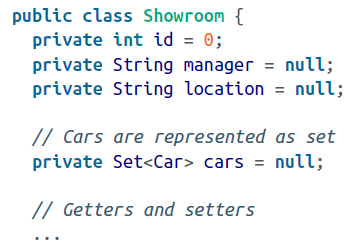
**Persisting Sets**

*Java.util.Set* represents an unordered data structure where duplicates are not allowed. Using sets is a straightforward just like lists. We’ll revisit the showroom cars example from the previous example to demonstrate how sets are used with Hibernate.

In our modified example, the collection of cars that belong to a showroom is modeled as *java.util.Set*; thus we define the cars variable as the Set type. We use *HashSet* as our concrete implementation of the *Set* interface.

The *Showroom* class is shown here:

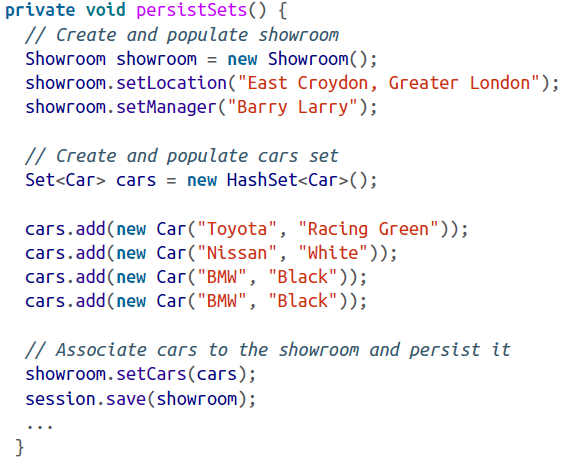


The notable change is the use of the *Set* collection instead of List. Once you finish modifying the *Showroom* class, the mapping of *Set* is done via the set tag, as demonstrated in the following snippet:



A Showroom instance is mapped to SHOWROOM\_SET table, whereas the cars variable representing the set Collection is mapped to the CARS\_SET table, as expected. The key element represents the presence of a foreign key in the CARS\_SET table. Hibernate adds this foreign key to the CARS\_SE table automatically. Hence, the CARS\_SET table, which is created and managed by Hibernate, will have the additional foreign key SHOWROOM\_ID, thus associating the two tables.

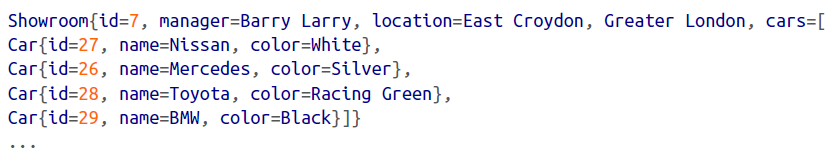
Create a test client as shown here:



In the preceding example, we created a *Showroom* object to which we’ve added three new cars.We are using *HashSet* as our concrete implementation for our *cars* collection. Did you notice that we are trying to add another BMW to the set? The set would identify these two cars as identical bases on equality matching and hence would throw away the duplicate one.

When working with sets, we need to satisfy an equality requirement: we must create *equals* and *hashCode* methods in the *Car* object. As we know, each individual item that’s being added to the set must be unique. The equals and *hashCode* methods would help to satisfy this requirement. Make sure the *equals* and *hashCode* contracts are fulfilled correctly—for example, use the fields that fields will identify a car uniquely.

The *retrieveSets* test method would fetch the persisted set from the database, as shown in this listing:



Did you notice that the BMW isn’t listed, although we added to the set earlier? This demonstrates the set’s “exclusion of duplicates” policy.