one-to-many as one-to-one

1. each entity has their own individual identities
2. each of them has their own independent existence

1..1

```
class Account {
  int accountNo;
  String accountHolderName;
  String accountType;
  String ifscCode;
  double balance;
  @OneToMany
  @JoinColumn(name="issued_account_no", unique="true")
  Set<Locker> assignedLockers;
}
```

1..1

```
class Locker {
  int lockerNo;
  String keyNo;
  String dimensions;
  double charges;
  @ManyToOne
  @JoinColumn(name="issued_account_no", unique="true")
  Account issuedAccount;
}
```

```
account
account_no (pk)
account_holder_nm
account_type
ifsc_code
balance
```

```
locker
locker_no (pk)
key_no
dimensions
charges
issued_account_no (fk)
                   (unique)
```

here each of the entities has their own
1. individual identities
2. independent existences

so as both the tables has their own pk columns, the only way to establish the relationship is by writing pk column of one table as foreign key in another table, but by doing so it becomes one-to-many. But from business point of view these has to act like one-to-one only, that can be achieved by making foreign key column as unique

since the relationship column is in another table, from entities world, it has to be still represented as one-to-many associate only. So inside the Account entity class we need to declare Set<Locker> assignedLockers wherein always the Set holds at max 1 Locker only because of 1:1 relationship in business.