Q1 Create a class Author with instance variables firstName, lastName and age.
Solution

```
package com.demo.hibernate;

import javax.persistence.Entity;
import javax.persistence.Id;

//added annotations to mark it as Hibernate Entity, required for Q2 onwards
@Entity
public class Author {
  @Id
  private Integer id;
  private String firstName;
  private String lastName;
  private Integer age;
  public Author(){}

 public Author(Integer id, String firstName, String lastName, Integer age) {
  this.id = id;
  this.firstName = firstName;
  this.lastName = lastName;
  this.age = age;
 }


  public Integer getId() {
     return id;
  }

  public void setId(Integer id) {
     this.id = id;
  }

  public String getFirstName() {
     return firstName;
  }

  public void setFirstName(String firstName) {
     this.firstName = firstName;
  }

  public String getLastName() {
     return lastName;
  }

  public void setLastName(String lastName) {
     this.lastName = lastName;
  }

  public Integer getAge() {
```

```
        return age;
    }

    public void setAge(Integer age) {
        this.age = age;
    }

    @Override
    public String toString() {
        return "Author{" +
                "id=" + id +
                ", firstName='" + firstName + '\'' +
                ", lastName='" + lastName + '\'' +
                ", age=" + age +
                '}';
    }
}
```

Q2. Perform CRUD operation for Author class.

Solution

Main.java

```
package com.demo.hibernate;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class Main {
  public static void main(String[] args) {

    Author author=new Author(1,"Surbhi","Garg",23);
    Author author2=new Author(2,"Shivam","Khanna",22);

    SessionFactory sessionFactory=new Configuration().configure().buildSessionFactory();

    Session session=sessionFactory.openSession();

    session.beginTransaction();
    //create
    session.save(author);
    session.save(author2);
    //read
    Author author1=session.get(Author.class,1);
    System.out.println("Author with id 1: "+author1);
    //update
    author1.setFirstName("Sakshi");
    session.update(author1);
    Author author3=session.get(Author.class,1);
    System.out.println("Author with id 1 after update: "+author3);
    //delete
    session.delete(author1);

    session.getTransaction().commit();
  }
}
```

Output

```
Hibernate: create table Author (id integer not null, age integer, firstName varchar(255), lastName varcha
Mar 19, 2019 12:20:33 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorN
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.
Mar 19, 2019 12:20:34 PM org.hibernate.tool.schema.internal.SchemaCreatorImpl applyImportSources
INFO: HHH000476: Executing import script 'org.hibernate.tool.schema.internal.exec.ScriptSourceInputNonEx
Author with id 1: Author{id=1, firstName='Surbhi', lastName='Garg', age=23}
Author with id 1 after update: Author{id=1, firstName='Sakshi', lastName='Garg', age=23}
Hibernate: insert into Author (age, firstName, lastName, id) values (?, ?, ?, ?)
Hibernate: insert into Author (age, firstName, lastName, id) values (?, ?, ?, ?)
Hibernate: delete from Author where id=?
```

Q3. Use hbm2ddl create to introduce Date of Birth for Author.

Solution

Author.java

```
package com.demo.hibernate;

import javax.persistence.Entity;
import javax.persistence.Id;
import java.util.Date;

//added annotations to mark it as Hibernate Entity, required for Q2 onwards
@Entity
public class Author {
  @Id
  private Integer id;
  private String firstName;
  private String lastName;
  private Integer age;
  //added field for question3
  private Date dob;

  public Date getDob() {
     return dob;
  }

  public void setDob(Date dob) {
     this.dob = dob;
  }
 @Override
  public String toString() {
     return "Author{" +
           "id=" + id +
           ", firstName='" + firstName + '\'' +
           ", lastName='" + lastName + '\'' +
           ", age=" + age +
           ", dob=" + dob +
           '}';
  }
}
```

Hibernate.cfg.xml

```
<property name="hbm2ddl.auto">create</property>
```

Main.java

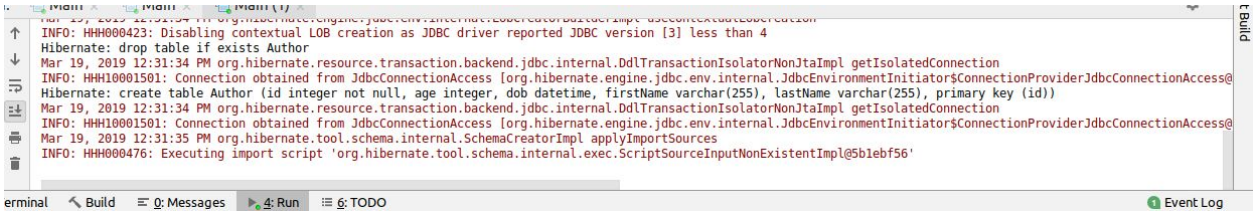
```
package com.demo.hibernate.question3;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class Main {
```

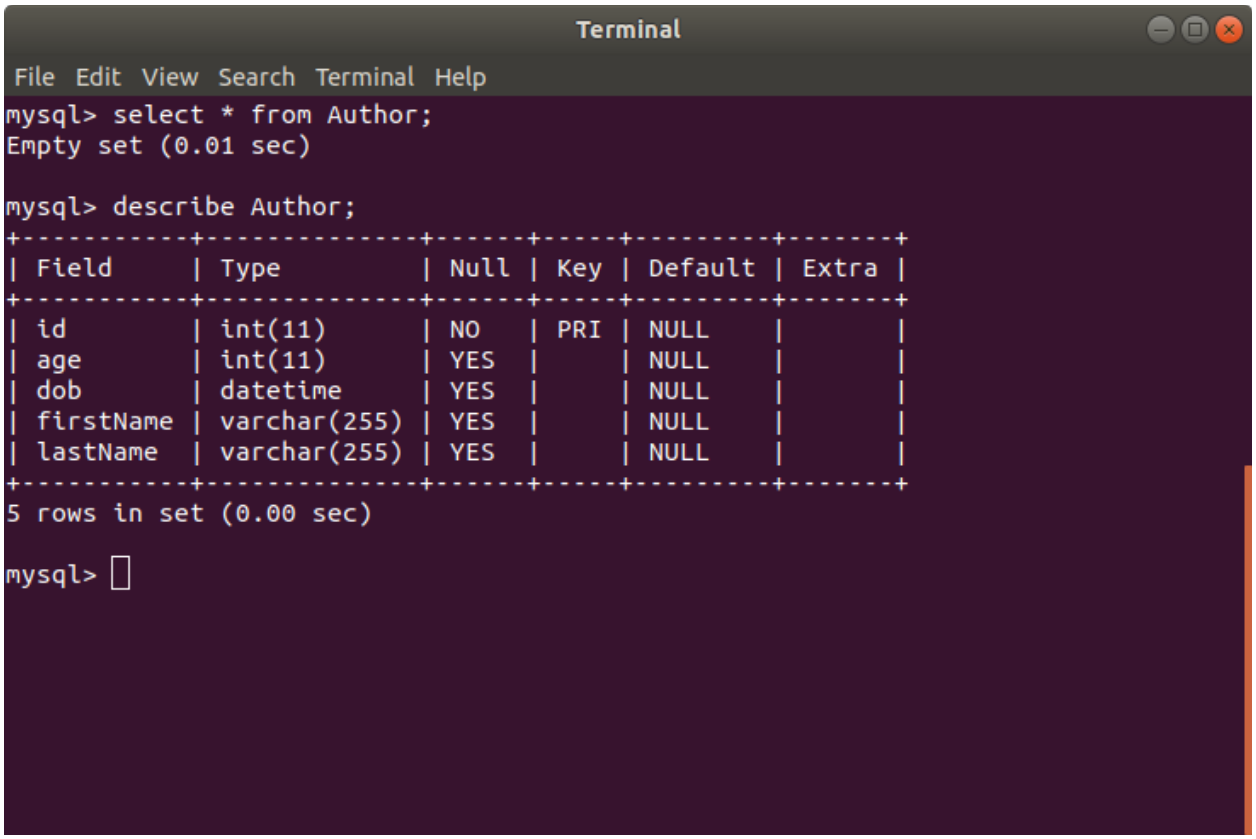
```
  public static void main(String[] args) {

      SessionFactory sessionFactory=new Configuration().configure().buildSessionFactory();

      Session session=sessionFactory.openSession();

      session.beginTransaction();

      session.getTransaction().commit();
  }
}
```

Output

Inference

First table is dropped, and a complete new table is created with added dob field, leading to losing all previous data, if any.





Q4. Use hbm2dll update to insert at least 4 records for Author.

Solution

Hibernate.cfg.xml

```
<property name="hbm2ddl.auto">update</property>
```

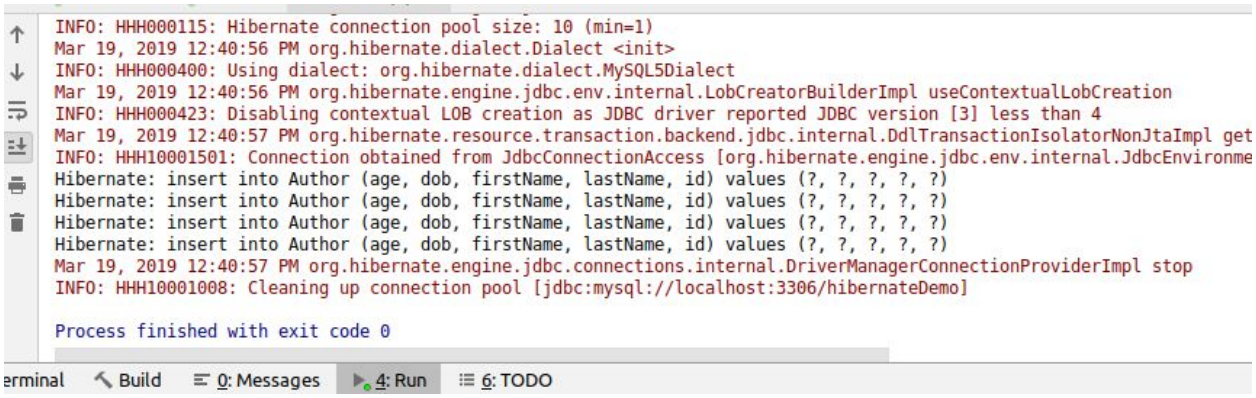
Main.java

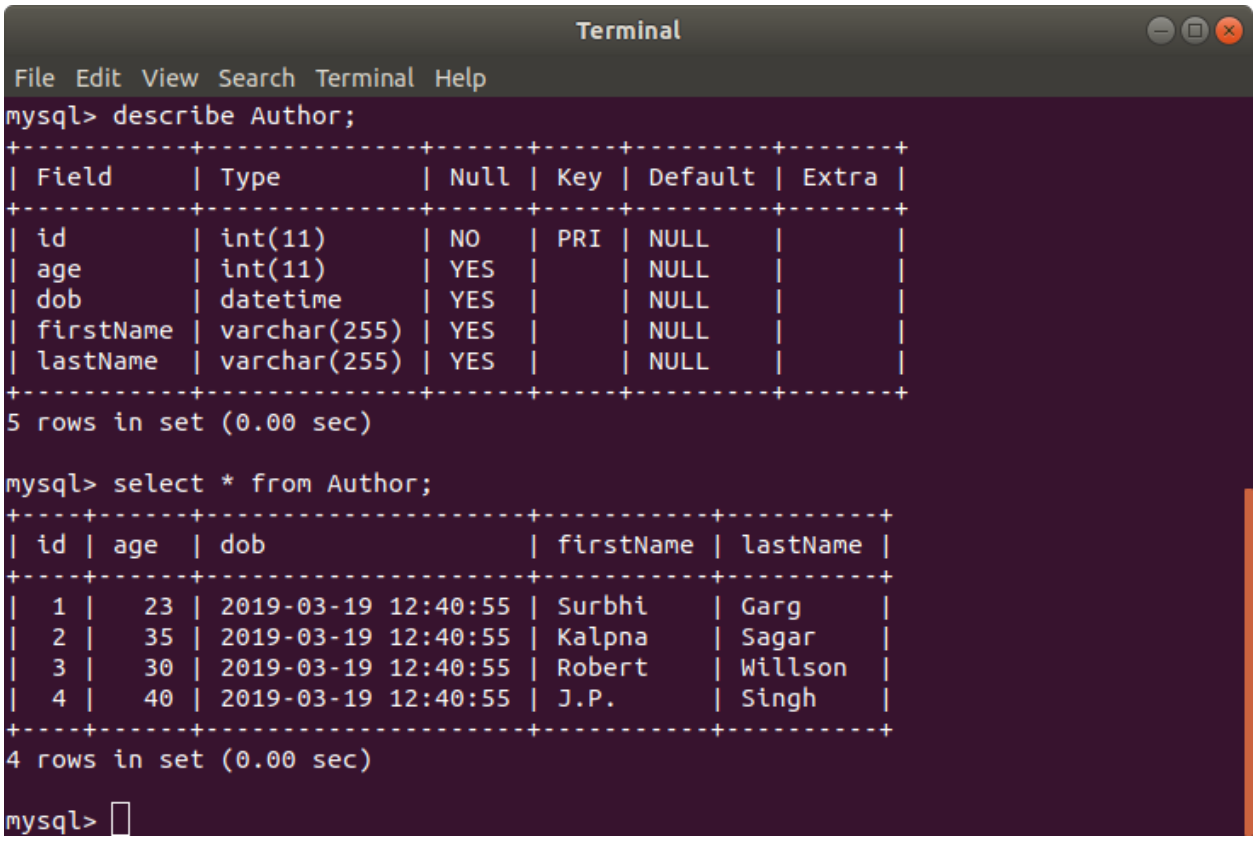
```
package com.demo.hibernate.question4;

import com.demo.hibernate.Author;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

import java.util.Date;

public class Main {
  public static void main(String[] args) {
     Author author=new Author(1,"Surbhi","Garg",23);
     author.setDob(new Date());

     Author author1=new Author(2,"Kalpna","Sagar",35);
     author1.setDob(new Date());

     Author author2=new Author(3,"Robert","Willson",30);
     author2.setDob(new Date());

     Author author3=new Author(4,"J.P.","Singh",40);
     author3.setDob(new Date());

     SessionFactory sessionFactory=new Configuration().configure().buildSessionFactory();

     Session session=sessionFactory.openSession();
     session.beginTransaction();

        session.save(author);
        session.save(author1);
        session.save(author2);
        session.save(author3);
     session.getTransaction().commit();
     session.close();
     sessionFactory.close();
  }
}
```

Output

Inference

Table is not dropped, as it was already created. Only insertion of records happens.





Q5. Perform hbm2dll create-drop by closing session factory.

Solution

Hibernate.cfg.xml

```
property name="hbm2ddl.auto">create-drop</property>
```

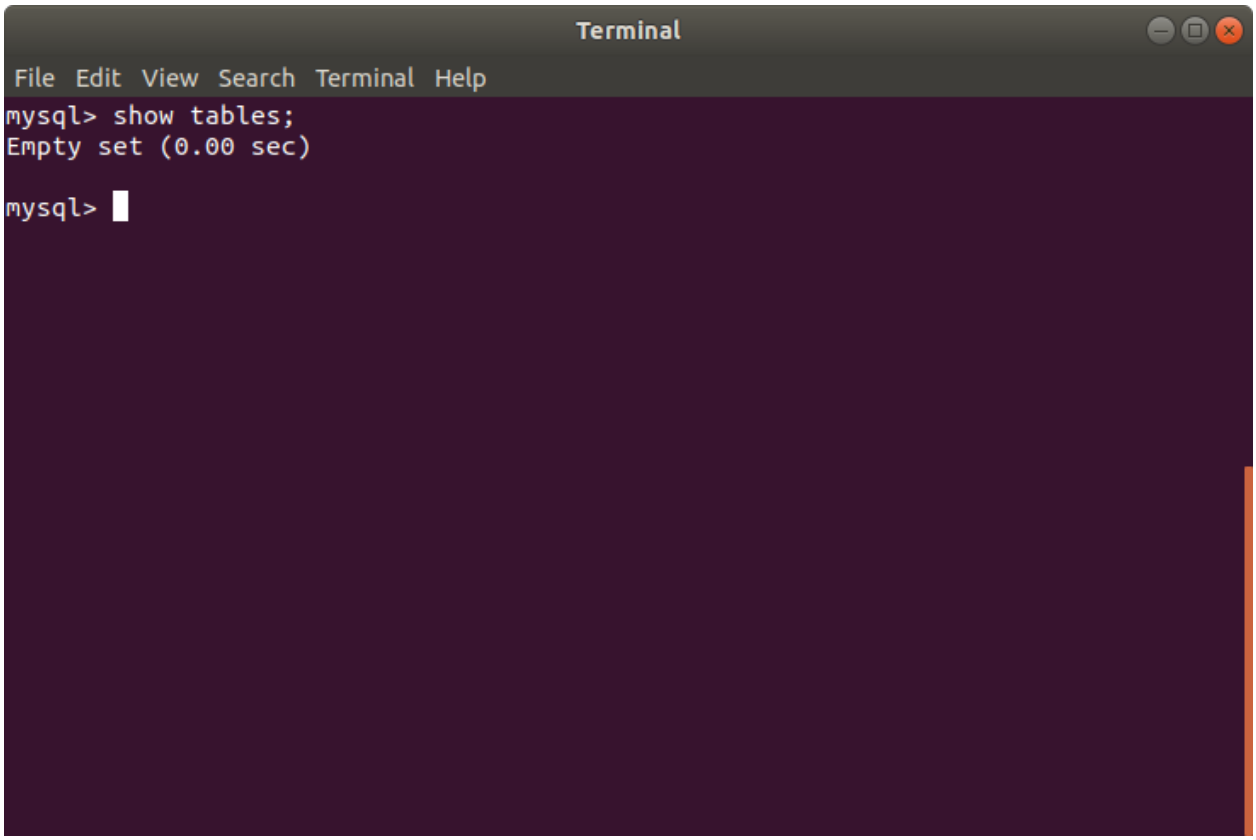
Main.java

```
package com.demo.hibernate.question5;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class Main {
  public static void main(String[] args) {

    SessionFactory sessionFactory=new Configuration().configure().buildSessionFactory();

    Session session=sessionFactory.openSession();

    session.beginTransaction();

    session.getTransaction().commit();
    session.close();
    sessionFactory.close();
  }
}
```

Output

Table is dropped as soon as sessionfactory is closed

```
Hibernate: drop table if exists Author
Mar 19, 2019 12:46:51 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcC
Hibernate: create table Author (id integer not null, age integer, dob datetime, firstName varchar(255), lastName varchar(255), primary key (id))
Mar 19, 2019 12:46:51 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcC
Mar 19, 2019 12:46:52 PM org.hibernate.tool.schema.internal.SchemaCreatorImpl applyImportSources
INFO: HHH000476: Executing import script 'org.hibernate.tool.schema.internal.exec.ScriptSourceInputNonExistentImpl@5b1ebf56'
Hibernate: drop table if exists Author
Mar 19, 2019 12:46:52 PM org.hibernate.tool.schema.internal.SchemaDropperImpl$DelayedDropActionImpl perform
INFO: HHH000477: Starting delayed drop of schema as part of SessionFactory shut-down'
Mar 19, 2019 12:46:52 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcC
Mar 19, 2019 12:46:52 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl stop
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost:3306/hibernateDemo]

Process finished with exit code 0
```



Q6. Rename all the fields using column annotation.

Solution

Author.java

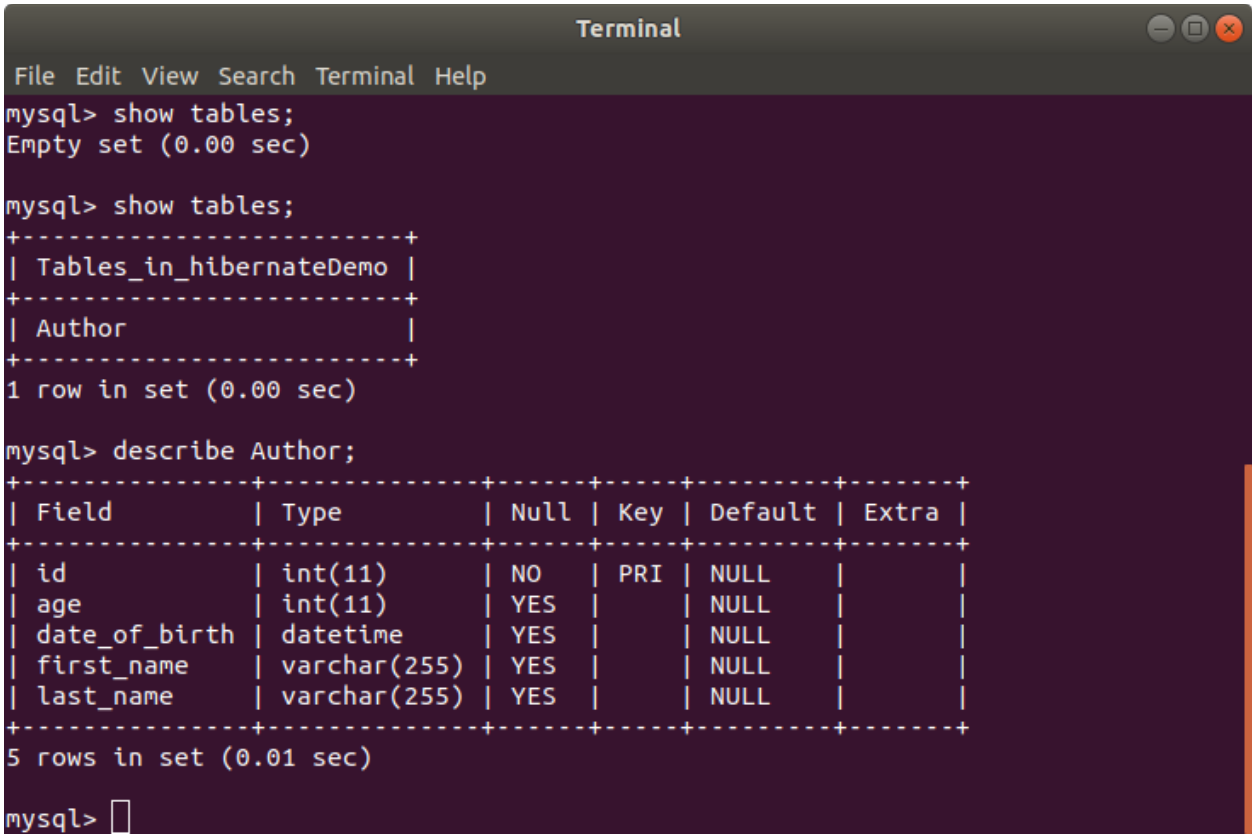
```
package com.demo.hibernate;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import java.util.Date;

//added annotations to mark it as Hibernate Entity, required for Q2 onwards
//added @Column for Question6
@Entity
public class Author {
  @Id
  @Column(name = "id")
  private Integer id;

  @Column(name="first_name")
  private String firstName;

  @Column(name = "last_name")
  private String lastName;

  @Column(name = "age")
  private Integer age;

  @Column(name="date_of_birth")
  //added field for question3
  private Date dob;

  public Date getDob() {
     return dob;
  }

  public void setDob(Date dob) {
     this.dob = dob;
  }

  public Author(){}

  public Author(Integer id, String firstName, String lastName, Integer age) {
     this.id = id;
     this.firstName = firstName;
     this.lastName = lastName;
     this.age = age;
  }

  public Integer getId() {
```

```
    return id;
  }

  public void setId(Integer id) {
    this.id = id;
  }

  public String getFirstName() {
    return firstName;
  }

  public void setFirstName(String firstName) {
    this.firstName = firstName;
  }

  public String getLastName() {
    return lastName;
  }

  public void setLastName(String lastName) {
    this.lastName = lastName;
  }

  public Integer getAge() {
    return age;
  }

  public void setAge(Integer age) {
    this.age = age;
  }

  @Override
  public String toString() {
    return "Author{" +
          "id=" + id +
          ", firstName='" + firstName + '\'' +
          ", lastName='" + lastName + '\'' +
          ", age=" + age +
          ", dob=" + dob +
          '}';
  }
}
```

Main.java

Run `com.demo.hibernate.question3.Main;`

Output


Terminal
File Edit View Search Terminal Help
mysql> show tables;
Empty set (0.00 sec)
mysql> show tables;
| Tables_in_hibernateDemo |
| Author |
1 row in set (0.00 sec)
mysql> describe Author;
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
age	int(11)	YES		NULL	
date_of_birth	datetime	YES		NULL	
first_name	varchar(255)	YES		NULL	
last_name	varchar(255)	YES		NULL	
5 rows in set (0.01 sec)
mysql>

Q7. Mark lastName as @Transient.

Author.java

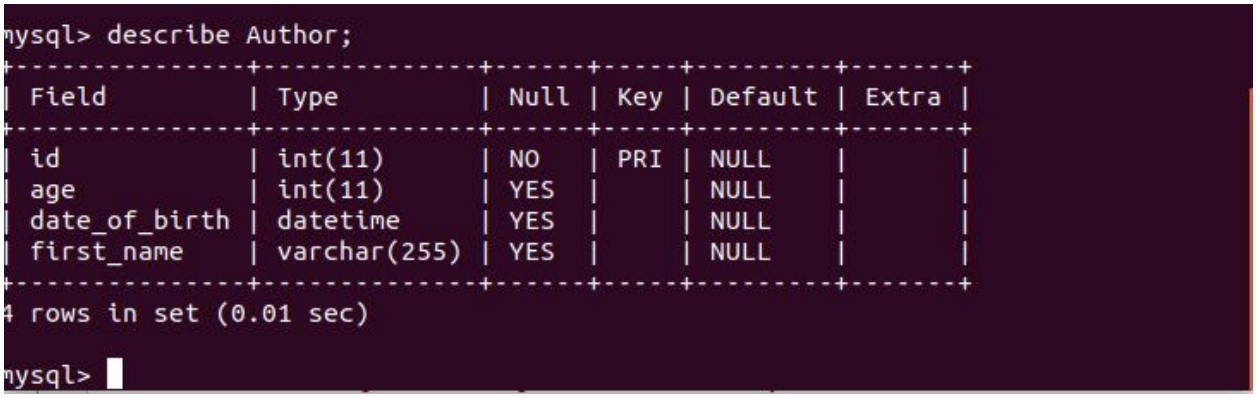
```
package com.demo.hibernate;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.Transient;
import java.util.Date;

//added annotations to mark it as Hibernate Entity, required for Q2 onwards
//added @Column for Question6
@Entity
public class Author {


  @Column(name = "last_name")
  //for Question7
  @Transient
  private String lastName;
}
```

Main.java

Run **com.demo.hibernate.question3.Main;**

Output

Field last_name is not created



Q8. Use @Temporal for date of birth of Author.

Solution

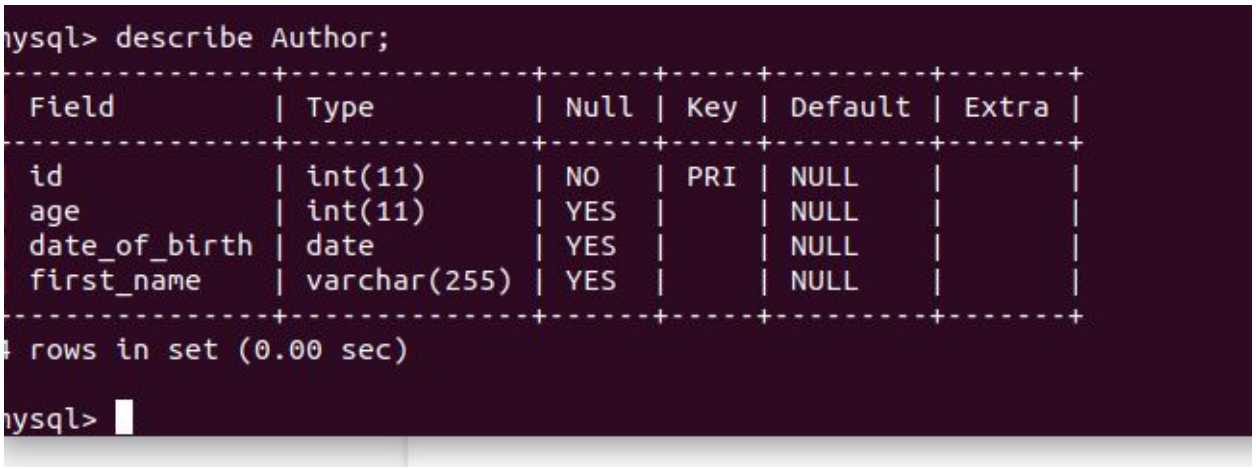
```
package com.demo.hibernate;

import javax.persistence.*;
import java.util.Date;

//added annotations to mark it as Hibernate Entity, required for Q2 onwards
//added @Column for Question6
@Entity
public class Author {
  @Column(name="date_of_birth")
  //added field for question3
  //forQuestion8-@Temporal
  @Temporal(TemporalType.DATE)
  private Date dob;
}
```

Main/java

Run **com.demo.hibernate.question3.Main;**

Output

Type of date_of_birth is changed to date, which was datetime earlier



Q9. Generate Id for Author Using IDENTITY and TABLE strategy.

Solution

Author.java

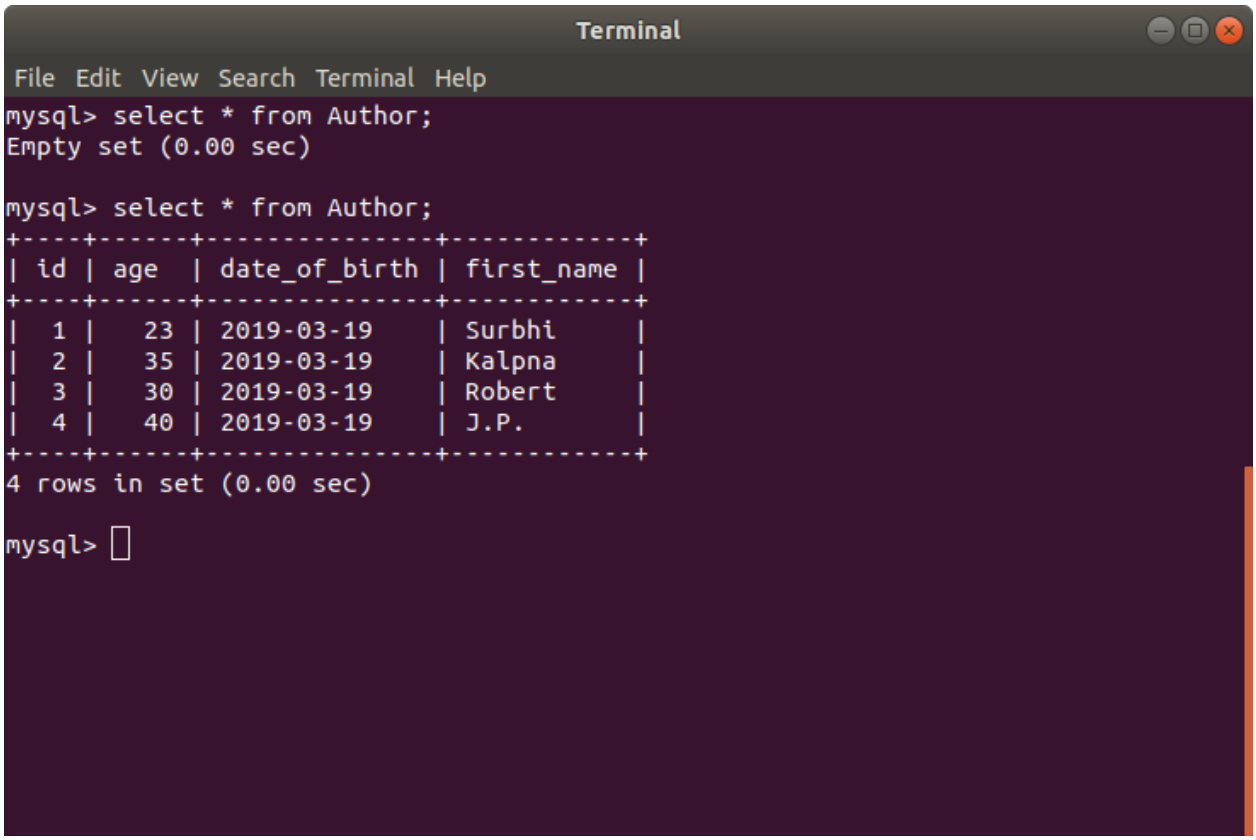
```
package com.demo.hibernate;

import javax.persistence.*;
import java.util.Date;

//added annotations to mark it as Hibernate Entity, required for Q2 onwards
//added @Column for Question6
@Entity
public class Author {
  @Id
  @Column(name = "id")
  //question9-->identity
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private Integer id;

  public Author(String firstName, String lastName, Integer age, Date dob) {
     this.firstName = firstName;
     this.lastName = lastName;
     this.age = age;
     this.dob = dob;
  }
}
```

Main.java

```
package com.demo.hibernate.question9;

import com.demo.hibernate.Author;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

import java.util.Date;

public class Main {
  public static void main(String[] args) {
  Author author=new Author("Surbhi","Garg",23,new Date());

  Author author1=new Author("Kalpna","Sagar",35,new Date());

  Author author2=new Author("Robert","Willson",30,new Date());

  Author author3=new Author("J.P.","Singh",40,new Date());

  SessionFactory sessionFactory=new Configuration().configure().buildSessionFactory();

  Session session=sessionFactory.openSession();
```

```
        session.beginTransaction();

            session.save(author);
            session.save(author1);
            session.save(author2);
            session.save(author3);
        session.getTransaction().commit();
        session.close();
        sessionFactory.close();
    }
}
```

Output



Using table strategy

Author.java

```
package com.demo.hibernate;

import javax.persistence.*;
import java.util.Date;

//added annotations to mark it as Hibernate Entity, required for Q2 onwards
//added @Column for Question6
@Entity
public class Author {
  @Id
  @Column(name = "id")
  //question9-->identity
  @GeneratedValue(strategy = GenerationType.TABLE)
  private Integer id;

  public Author(String firstName, String lastName, Integer age, Date dob) {
     this.firstName = firstName;
     this.lastName = lastName;
     this.age = age;
     this.dob = dob;
  }
}
```

Output

Id is generated on the basis of a centralized table hibernate_sequence

```
Hibernate: select tbl.next_val from hibernate_sequences tbl where tbl.sequence_name=? for update
Hibernate: insert into hibernate_sequences (sequence_name, next_val)  values (?,?)
Hibernate: update hibernate_sequences set next_val=?  where next_val=? and sequence_name=?
Hibernate: select tbl.next_val from hibernate_sequences tbl where tbl.sequence_name=? for update
Hibernate: update hibernate_sequences set next_val=?  where next_val=? and sequence_name=?
Hibernate: select tbl.next_val from hibernate_sequences tbl where tbl.sequence_name=? for update
Hibernate: update hibernate_sequences set next_val=?  where next_val=? and sequence_name=?
Hibernate: select tbl.next_val from hibernate_sequences tbl where tbl.sequence_name=? for update
Hibernate: update hibernate_sequences set next_val=?  where next_val=? and sequence_name=?
Hibernate: insert into Author (age, date_of_birth, first_name, id) values (?, ?, ?, ?)
Hibernate: insert into Author (age, date_of_birth, first_name, id) values (?, ?, ?, ?)
Hibernate: insert into Author (age, date_of_birth, first_name, id) values (?, ?, ?, ?)
Hibernate: insert into Author (age, date_of_birth, first_name, id) values (?, ?, ?, ?)
Mar 19, 2019 1:23:49 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl stop
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost:3306/hibernateDemo]

Process finished with exit code 0
```

Terminal

File Edit View Search Terminal Help

```
| Tables_in_hibernateDemo |
+-------------------------+
| Author                  |
| hibernate_sequences     |
+-------------------------+
2 rows in set (0.00 sec)

mysql> select * from Author;
+----+------+---------------+------------+
| id | age  | date_of_birth | first_name |
+----+------+---------------+------------+
|  1 |   23 | 2019-03-19    | Surbhi     |
|  2 |   35 | 2019-03-19    | Kalpna     |
|  3 |   30 | 2019-03-19    | Robert     |
|  4 |   40 | 2019-03-19    | J.P.       |
+----+------+---------------+------------+
4 rows in set (0.00 sec)

mysql> select * from hibernate_sequences;
+---------------+----------+
| sequence_name | next_val |
+---------------+----------+
| default       |        5 |
+---------------+----------+
```

Q10. Create a class Address for Author with instance variables streetNumber, location, State.

Address.java

```
package com.demo.hibernate;

import javax.persistence.Embeddable;
//for Question11 added annotation
@Embeddable
public class Address {
  private int streetNumber;
  private String location;
  private String state;

  @Override
  public String toString() {
    return "Address{" +
        "streetNumber=" + streetNumber +
        ", location='" + location + '\'' +
        ", state='" + state + '\'' +
        '}';
  }

  public int getStreetNumber() {
    return streetNumber;
  }

  public void setStreetNumber(int streetNumber) {
    this.streetNumber = streetNumber;
  }

  public String getLocation() {
    return location;
  }

  public void setLocation(String location) {
    this.location = location;
  }

  public String getState() {
    return state;
  }

  public void setState(String state) {
    this.state = state;
  }
}
```

Q11. Create instance variable of Address class inside Author class and save it as embedded object.

Solution

Author.java

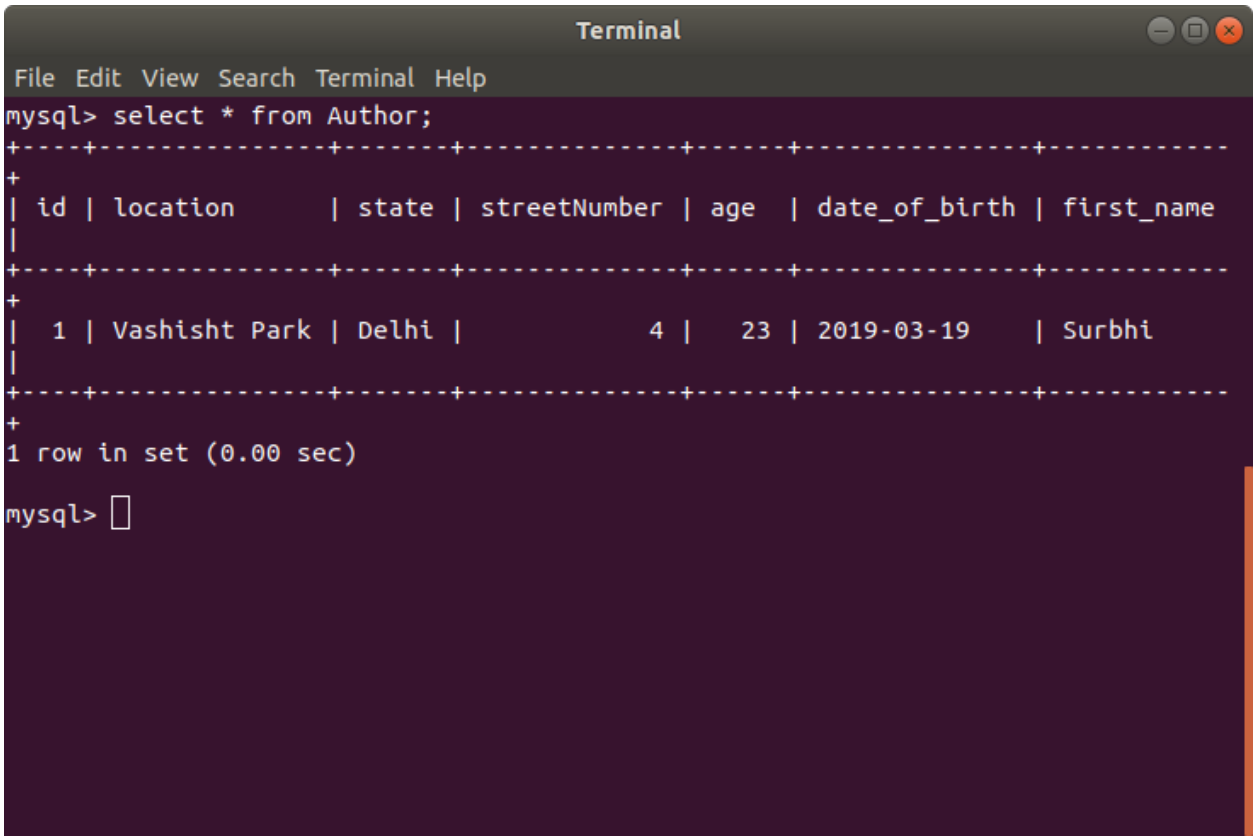
```
package com.demo.hibernate;

import javax.persistence.*;
import java.util.Date;

//added annotations to mark it as Hibernate Entity, required for Q2 onwards
//added @Column for Question6
@Entity
public class Author {
  //Question11
  @Embedded
  Address address;

  public Address getAddress() {
     return address;
  }

  public void setAddress(Address address) {
     this.address = address;
  }


}
```

Main.java

```
package com.demo.hibernate.question11;

import com.demo.hibernate.Address;
import com.demo.hibernate.Author;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

import java.util.Date;

public class Main {
  public static void main(String[] args) {
     Author author=new Author("Surbhi","Garg",23,new Date());

     Address address=new Address();
     address.setLocation("Vashisht Park");
     address.setState("Delhi");
     address.setStreetNumber(4);

     author.setAddress(address);
```

```
    SessionFactory sessionFactory=new Configuration().configure().buildSessionFactory();

    Session session=sessionFactory.openSession();

    session.beginTransaction();

       session.save(author);

    session.getTransaction().commit();
  }
}
```

Output



Q12. Introduce a List of subjects for author.

Solution

Author.java

```
package com.demo.hibernate;

import javax.persistence.*;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

//added annotations to mark it as Hibernate Entity, required for Q2 onwards
//added @Column for Question6
@Entity
public class Author {

  //Question12 and 13
  @ElementCollection
  List<String>subjectList=new ArrayList<String>();
}
```

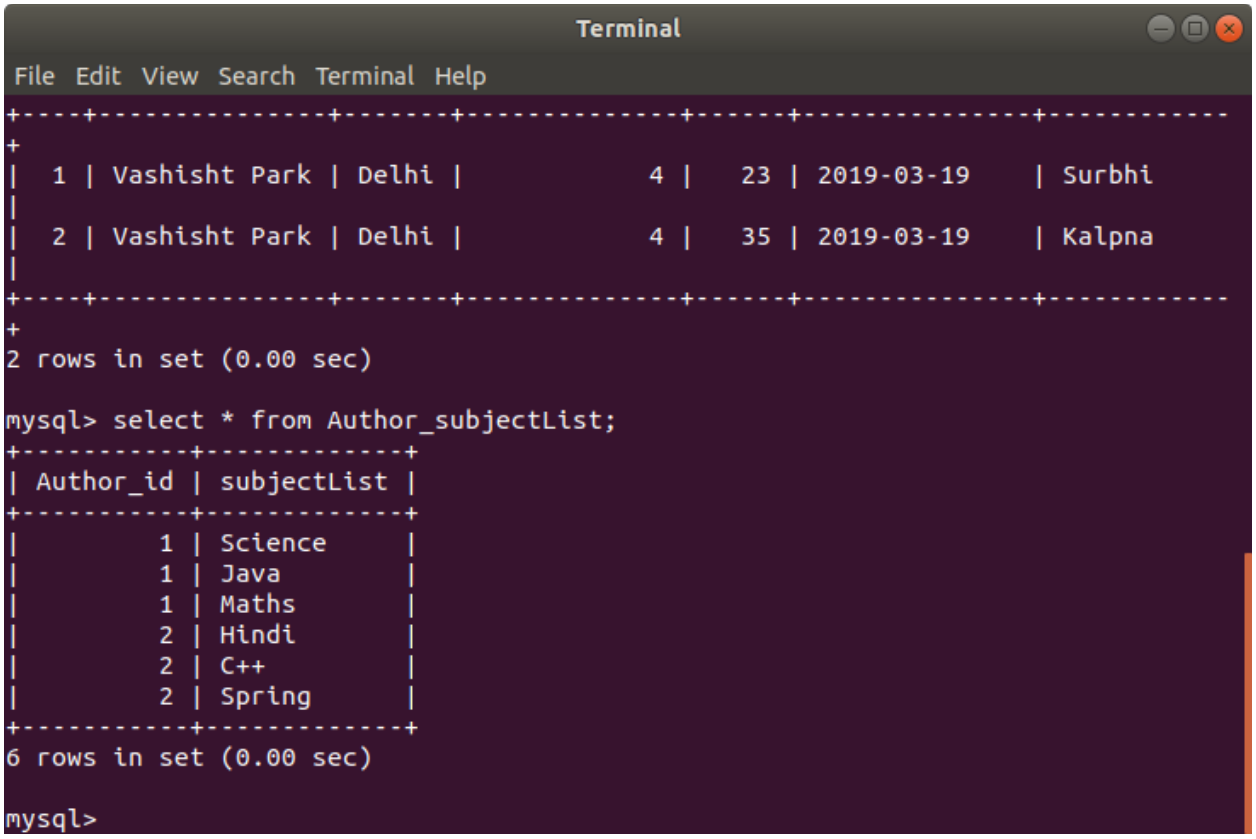
Q13. Persist 3 subjects for each author.

Solution

Main.java

```
package com.demo.hibernate.question13;

import com.demo.hibernate.Address;
import com.demo.hibernate.Author;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

import java.util.Arrays;
import java.util.Date;

public class Main {
  public static void main(String[] args) {

      Address address=new Address();
      address.setLocation("Vashisht Park");
      address.setState("Delhi");
      address.setStreetNumber(4);

      Author author=new Author("Surbhi","Garg",23,new Date());
      author.setAddress(address);
      author.setSubjectList(Arrays.asList("Science","Java","Maths"));

      Author author1=new Author("Kalpna","Sagar",35,new Date());
      author1.setAddress(address);
      author1.setSubjectList(Arrays.asList("Hindi","C++","Spring"));

      SessionFactory sessionFactory=new Configuration().configure().buildSessionFactory();

      Session session=sessionFactory.openSession();
      session.beginTransaction();

      session.save(author);
      session.save(author1);

      session.getTransaction().commit();
      session.close();
      sessionFactory.close();
  }
}
```

Output


Terminal
File Edit View Search Terminal Help
+----+---------------+-------+--------------+------+---------------+------------
+
| 1 | Vashisht Park | Delhi | 4 | 23 | 2019-03-19 | Surbhi
|
| 2 | Vashisht Park | Delhi | 4 | 35 | 2019-03-19 | Kalpna
|
+----+---------------+-------+--------------+------+---------------+------------
+
2 rows in set (0.00 sec)
mysql> select * from Author_subjectList;
+-----------+-------------+
| Author_id | subjectList |
+-----------+-------------+
1	Science
1	Java
1	Maths
2	Hindi
2	C++
2	Spring
+-----------+-------------+
6 rows in set (0.00 sec)
mysql>

Q14. Create an Entity book with an instance variable bookName.

Solution

Book.java

```
package com.demo.hibernate;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Book {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private Integer id;
  private String bookName;

  public String getBookName() {
    return bookName;
  }

  public void setBookName(String bookName) {
    this.bookName = bookName;
  }

  public Integer getId() {
    return id;
  }

@Override
public String toString() {
  return "Book{" +
        "id=" + id +
        ", bookName='" + bookName + '\'' +
        '}';
}

}
```

Q15. Implement One to One mapping between Author and Book.

Solution

Hibernate.cfg.xml

```
<mapping class="com.demo.hibernate.Book"/>
```

Author.java

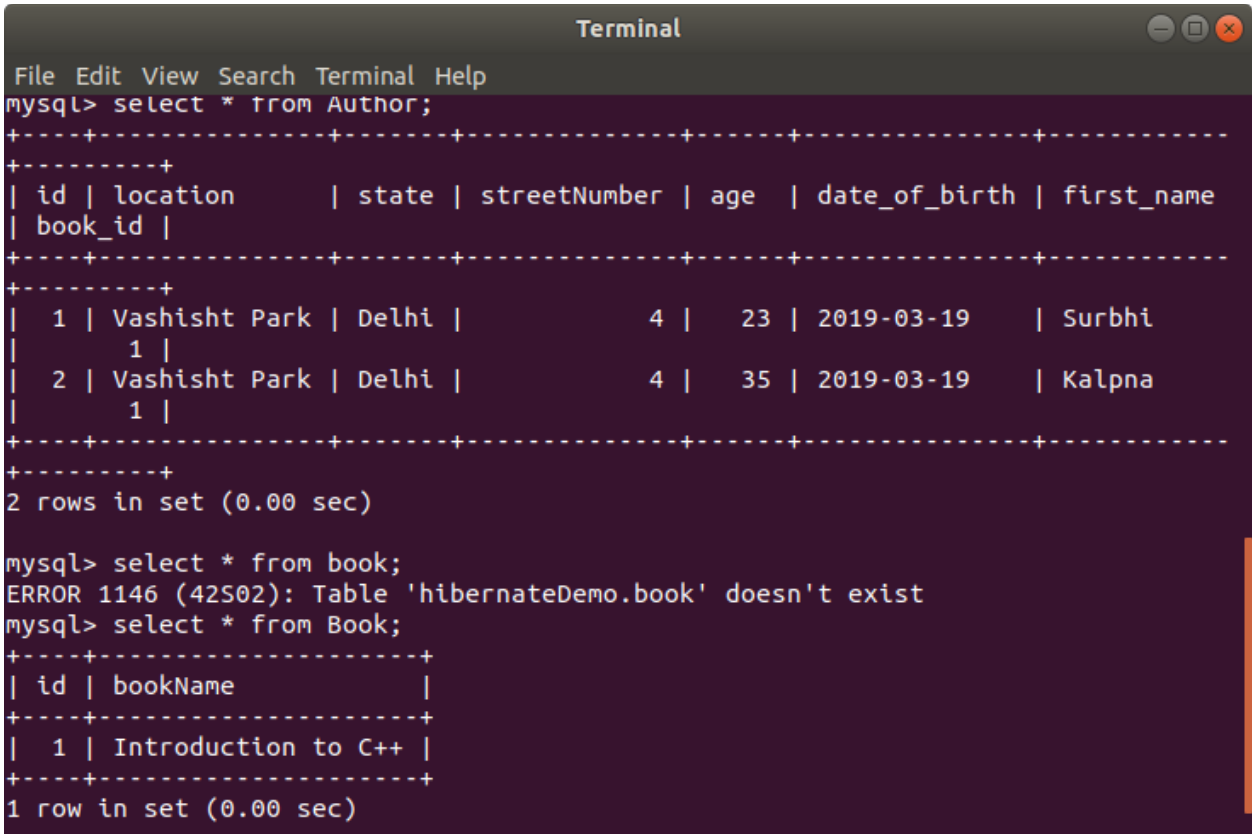
```
package com.demo.hibernate;

import javax.persistence.*;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

//added annotations to mark it as Hibernate Entity, required for Q2 onwards
//added @Column for Question6
@Entity
public class Author {
  @Embedded
  Address address;

  //Question12 and 13
  @ElementCollection
  List<String>subjectList=new ArrayList<String>();

  //Question15
  @OneToOne
  @JoinColumn(name = "book_id")
  Book book;

  public Book getBook() {
     return book;
  }

  public void setBook(Book book) {
     this.book = book;
  }
}
```

Main.java

```
package com.demo.hibernate.question15;

import com.demo.hibernate.Address;
import com.demo.hibernate.Author;
import com.demo.hibernate.Book;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

import java.util.Arrays;
```

```
import java.util.Date;

public class Main {
  public static void main(String[] args) {
     Book book=new Book();
     book.setBookName("Introduction to C++");

     Address address=new Address();
     address.setLocation("Vashisht Park");
     address.setState("Delhi");
     address.setStreetNumber(4);

     Author author=new Author("Surbhi","Garg",23,new Date());
     author.setAddress(address);
     author.setSubjectList(Arrays.asList("Science","Java","Maths"));
     author.setBook(book);

     Author author1=new Author("Kalpna","Sagar",35,new Date());
     author1.setAddress(address);
     author1.setSubjectList(Arrays.asList("Hindi","C++","Spring"));
     author1.setBook(book);

     SessionFactory sessionFactory=new Configuration().configure().buildSessionFactory();

     Session session=sessionFactory.openSession();
     session.beginTransaction();

     session.save(book);

     session.save(author);
     session.save(author1);

     session.getTransaction().commit();
     session.close();
     sessionFactory.close();
  }
}
```

Output


Terminal
File Edit View Search Terminal Help
mysql> select * from Author;
+----+--------------+-------+--------------+------+--------------+------------
+---------+
| id | location | state | streetNumber | age | date_of_birth | first_name
| book_id |
+----+--------------+-------+--------------+------+--------------+------------
+---------+
| 1 | Vashisht Park | Delhi | 4 | 23 | 2019-03-19 | Surbhi
| 1 |
| 2 | Vashisht Park | Delhi | 4 | 35 | 2019-03-19 | Kalpna
| 1 |
+----+--------------+-------+--------------+------+--------------+------------
+---------+
2 rows in set (0.00 sec)
mysql> select * from book;
ERROR 1146 (42S02): Table 'hibernateDemo.book' doesn't exist
mysql> select * from Book;
+----+---------------------+
| id | bookName |
+----+---------------------+
| 1 | Introduction to C++ |
+----+---------------------+
1 row in set (0.00 sec)

Q16. Implement One to Many Mapping between Author and Book(Unidirectional, BiDirectional and without additional table) and implement cascade save.

Solution

Unidirectional

Hibernate.cfg.xml

```
<mapping class="com.demo.hibernate.question16.unidirectional.Author"/>
```

Author.java

```
package com.demo.hibernate.question16.unidirectional;


import com.demo.hibernate.Address;
import com.demo.hibernate.Book;

import javax.persistence.*;
import java.util.*;


@Entity
@Table(name = "author1")
public class Author {
  @Id
  @Column(name = "id")
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private Integer id;

  @Column(name = "first_name")
  private String firstName;

  @Column(name = "last_name")
  @Transient
  private String lastName;

  @Column(name = "age")
  private Integer age;

  @Column(name = "date_of_birth")
  @Temporal(TemporalType.DATE)
  private Date dob;


  @Embedded
  Address address;

  @ElementCollection
  List<String> subjectList = new ArrayList<String>();
```

```
//Question16-->oneToMany--unidrectional
@OneToMany
@JoinTable(joinColumns = @JoinColumn(name = "auhor_id"),
      inverseJoinColumns = @JoinColumn(name = "book_id"))
Set<Book> bookSet = new HashSet<Book>();

public Author() {
}

public Author(String firstName, String lastName, Integer age, Date dob) {
   this.firstName = firstName;
   this.lastName = lastName;
   this.age = age;
   this.dob = dob;
}

public Author(Integer id, String firstName, String lastName, Integer age) {
   this.id = id;
   this.firstName = firstName;
   this.lastName = lastName;
   this.age = age;
}

public Address getAddress() {
   return address;
}

public void setAddress(Address address) {
   this.address = address;
}

public Date getDob() {
   return dob;
}

public void setDob(Date dob) {
   this.dob = dob;
}

public Integer getId() {
   return id;
}

public void setId(Integer id) {
   this.id = id;
}

public String getFirstName() {
   return firstName;
}
```

```
public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public String getLastName() {
    return lastName;
}

public void setLastName(String lastName) {
    this.lastName = lastName;
}

public Integer getAge() {
    return age;
}

public void setAge(Integer age) {
    this.age = age;
}

public List<String> getSubjectList() {
    return subjectList;
}

public void setSubjectList(List<String> subjectList) {
    this.subjectList = subjectList;
}

public Set<Book> getBookSet() {
    return bookSet;
}

public void setBookSet(Set<Book> bookSet) {
    this.bookSet = bookSet;
}

@Override
public String toString() {
    return "Author{" +
            "id=" + id +
            ", firstName='" + firstName + '\'' +
            ", lastName='" + lastName + '\'' +
            ", age=" + age +
            ", dob=" + dob +
            ", address=" + address +
            ", subjectList=" + subjectList +
            ", bookSet=" + bookSet +
            '}';
}
```

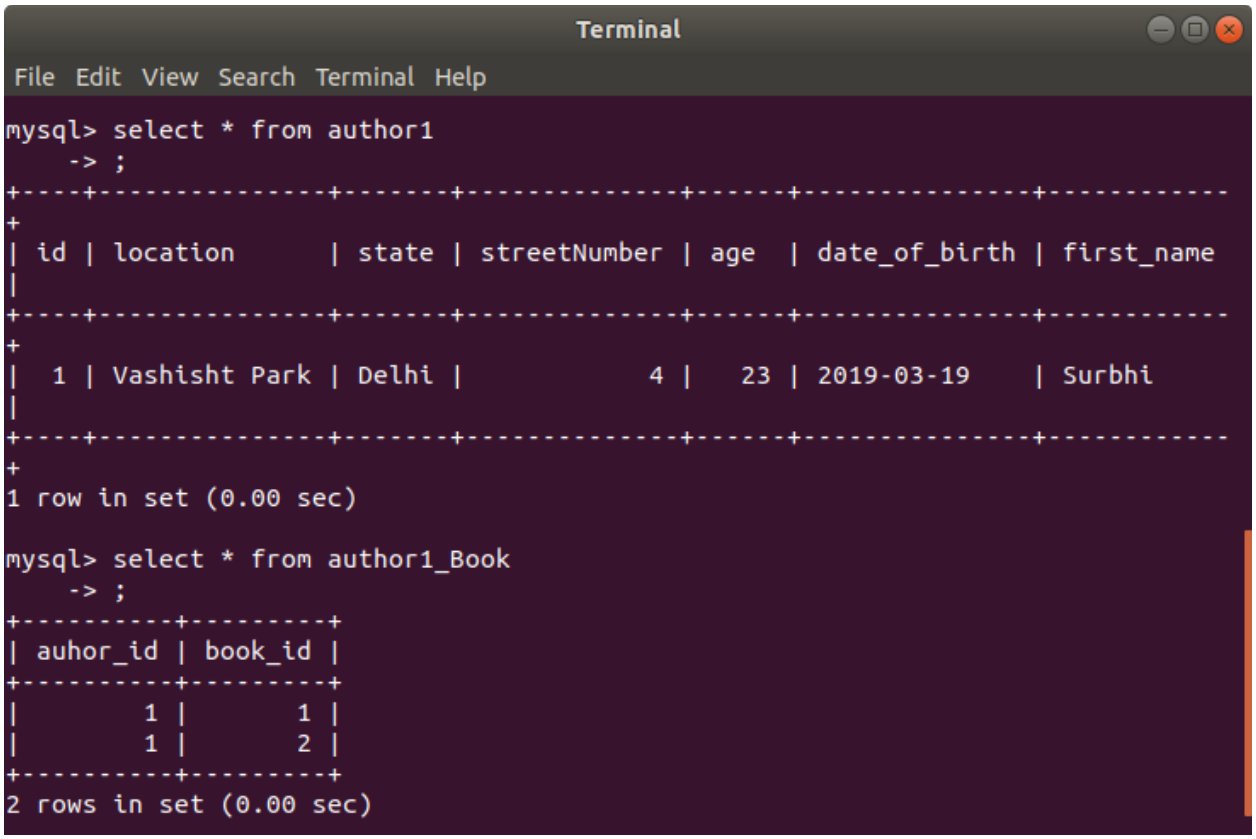
```
}
```

Main.java

```
package com.demo.hibernate.question16.unidirectional;

import com.demo.hibernate.Address;
import com.demo.hibernate.Book;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

import java.util.Arrays;
import java.util.Date;

public class Main {
  public static void main(String[] args) {

      Address address=new Address();
      address.setLocation("Vashisht Park");
      address.setState("Delhi");
      address.setStreetNumber(4);

      Author author=new Author(Surbhi","Garg",23,new Date());
      author.setAddress(address);
      author.setSubjectList(Arrays.asList("Science","Java","Maths"));

      Book book1=new Book();
      book1.setBookName("Introduction to C++");

      Book book2=new Book();
      book2.setBookName("Introdunction to java");


      author.getBookSet().addAll(Arrays.asList(book1,book2));


      SessionFactory sessionFactory=new Configuration().configure().buildSessionFactory();

      Session session=sessionFactory.openSession();
      session.beginTransaction();

      session.save(book1);
      session.save(book2);

      session.save(author);

      session.getTransaction().commit();
      session.close();
      sessionFactory.close();
  }
```

}

Output


Terminal
File Edit View Search Terminal Help
mysql> select * from author1
-> ;
+----+---------------+-------+--------------+------+---------------+------------
+
| id | location | state | streetNumber | age | date_of_birth | first_name
|
+----+---------------+-------+--------------+------+---------------+------------
+
| 1 | Vashisht Park | Delhi | 4 | 23 | 2019-03-19 | Surbhi
|
+----+---------------+-------+--------------+------+---------------+------------
+
1 row in set (0.00 sec)
mysql> select * from author1_Book
-> ;
+----------+---------+
| auhor_id | book_id |
+----------+---------+
| 1 | 1 |
| 1 | 2 |
+----------+---------+
2 rows in set (0.00 sec)

BiDirectional

Hibernate.cfg.xml

```
<mapping class="com.demo.hibernate.question16.bidirectional.Book"/>
<mapping class="com.demo.hibernate.question16.bidirectional.Author"/>
```

Author.java

```
package com.demo.hibernate.question16.bidirectional;

import com.demo.hibernate.Address;
import javax.persistence.*;
import java.util.*;


@Entity
@Table(name = "author2")
public class Author {
  @Id
  @Column(name = "id")
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private Integer id;

  @Column(name = "first_name")
  private String firstName;

  @Column(name = "last_name")
  @Transient
  private String lastName;

  @Column(name = "age")
  private Integer age;

  @Column(name = "date_of_birth")
  @Temporal(TemporalType.DATE)
  private Date dob;


  @Embedded
  Address address;

  @ElementCollection
  List<String> subjectList = new ArrayList<String>();

  //Question16-->oneToMany--unidrectional
  @OneToMany
  @JoinTable(joinColumns = @JoinColumn(name = "auhor_id"),
        inverseJoinColumns = @JoinColumn(name = "book_id"))
  Set<Book> bookSet = new HashSet<Book>();

  public Author() {
```

```
}

public Author(String firstName, String lastName, Integer age, Date dob) {
    this.firstName = firstName;
    this.lastName = lastName;
    this.age = age;
    this.dob = dob;
}

public Author(Integer id, String firstName, String lastName, Integer age) {
    this.id = id;
    this.firstName = firstName;
    this.lastName = lastName;
    this.age = age;
}

public Address getAddress() {
    return address;
}

public void setAddress(Address address) {
    this.address = address;
}

public Date getDob() {
    return dob;
}

public void setDob(Date dob) {
    this.dob = dob;
}

public Integer getId() {
    return id;
}

public void setId(Integer id) {
    this.id = id;
}

public String getFirstName() {
    return firstName;
}

public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public String getLastName() {
    return lastName;
```

```
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public Integer getAge() {
        return age;
    }

    public void setAge(Integer age) {
        this.age = age;
    }

    public List<String> getSubjectList() {
        return subjectList;
    }

    public void setSubjectList(List<String> subjectList) {
        this.subjectList = subjectList;
    }

    public Set<Book> getBookSet() {
        return bookSet;
    }

    public void setBookSet(Set<Book> bookSet) {
        this.bookSet = bookSet;
    }

    @Override
    public String toString() {
        return "Author{" +
                "id=" + id +
                ", firstName='" + firstName + '\'' +
                ", lastName='" + lastName + '\'' +
                ", age=" + age +
                ", dob=" + dob +
                ", address=" + address +
                ", subjectList=" + subjectList +
                ", bookSet=" + bookSet +
                '}';
    }
}
```

Book.java

```
package com.demo.hibernate.question16.bidirectional;
import javax.persistence.*;

@Entity
```

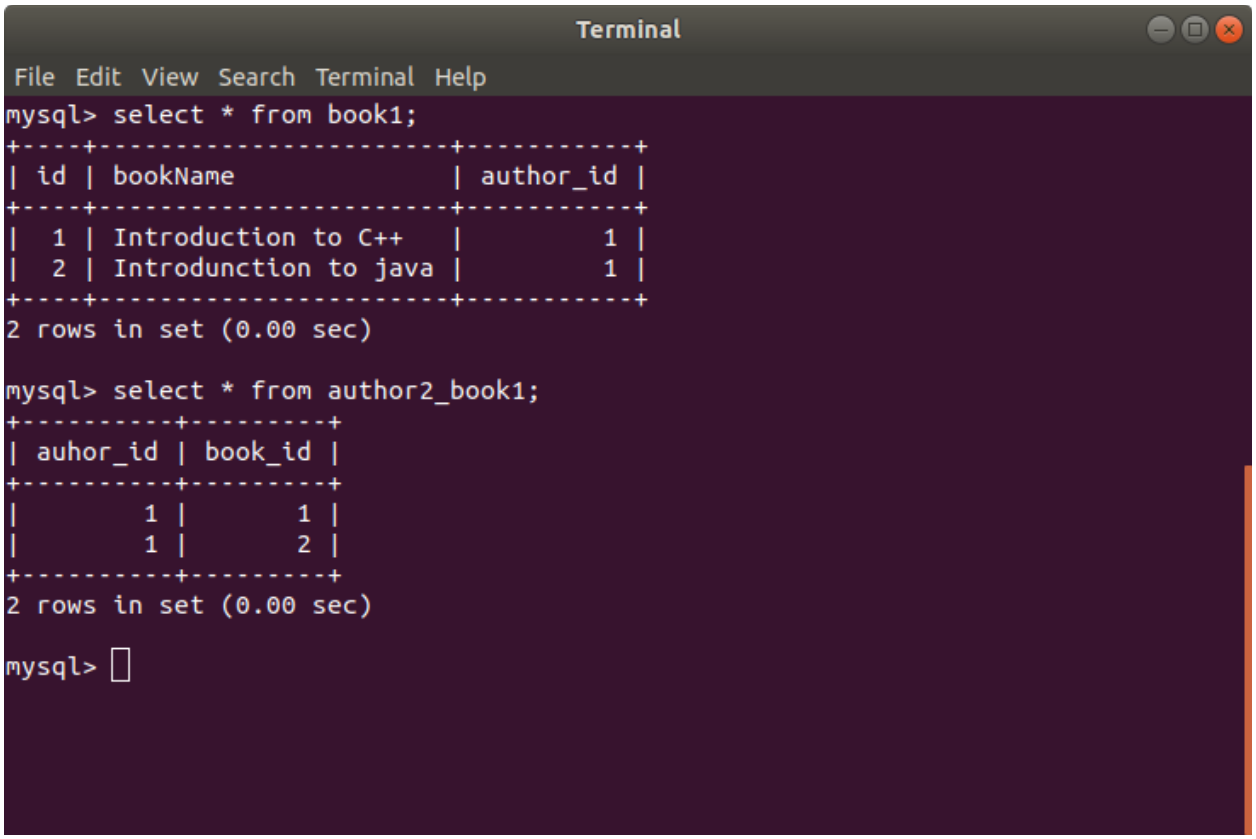
```java
@Table(name = "book1")
public class Book {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private Integer id;
  private String bookName;
  @ManyToOne
  Author author;

  public String getBookName() {
    return bookName;
  }

  public Author getAuthor() {
    return author;
  }

  public void setAuthor(Author author) {
    this.author = author;
  }

  public void setBookName(String bookName) {
    this.bookName = bookName;
  }

  public Integer getId() {
    return id;
  }

  @Override
  public String toString() {
    return "Book{" +
        "id=" + id +
        ", bookName='" + bookName + '\'' +
        '}';
  }
}
```

Main.java

```java
package com.demo.hibernate.question16.bidirectional;
import com.demo.hibernate.Address;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

import java.util.Arrays;
import java.util.Date;

public class Main {
  public static void main(String[] args) {
```

```
        Address address=new Address();
        address.setLocation("Vashisht Park");
        address.setState("Delhi");
        address.setStreetNumber(4);

        Author author=new Author("Surbhi","Garg",23,new Date());
        author.setAddress(address);
        author.setSubjectList(Arrays.asList("Science","Java","Maths"));

        Book book1=new Book();
        book1.setBookName("Introduction to C++");
        book1.setAuthor(author);


        Book book2=new Book();
        book2.setBookName("Introdunction to java");
        book2.setAuthor(author);

        author.getBookSet().addAll(Arrays.asList(book1,book2));


        SessionFactory sessionFactory=new Configuration().configure().buildSessionFactory();

        Session session=sessionFactory.openSession();
        session.beginTransaction();

        session.save(author);
        session.save(book1);
        session.save(book2);

        session.getTransaction().commit();
        session.close();
        sessionFactory.close();
    }
}
```

Output


Terminal
File Edit View Search Terminal Help
mysql> select * from book1;
+----+-----------------------+-----------+
| id | bookName | author_id |
+----+-----------------------+-----------+
| 1 | Introduction to C++ | 1 |
| 2 | Introdunction to java | 1 |
+----+-----------------------+-----------+
2 rows in set (0.00 sec)
mysql> select * from author2_book1;
+----------+---------+
| auhor_id | book_id |
+----------+---------+
| 1 | 1 |
| 1 | 2 |
+----------+---------+
2 rows in set (0.00 sec)
mysql>

Without Additional table

Hibernate.cfg.xml

```
<mapping class="com.demo.hibernate.question16.noaddtionaltable.Book"/>
<mapping class="com.demo.hibernate.question16.noaddtionaltable.Author"/>
```

Author.java

```
package com.demo.hibernate.question16.noaddtionaltable;

import com.demo.hibernate.Address;

import javax.persistence.*;
import java.util.*;


@Entity
@Table(name = "author3")
public class Author {
  @Id
  @Column(name = "id")
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private Integer id;

  @Column(name = "first_name")
  private String firstName;

  @Column(name = "last_name")
  @Transient
  private String lastName;

  @Column(name = "age")
  private Integer age;

  @Column(name = "date_of_birth")
  @Temporal(TemporalType.DATE)
  private Date dob;


  @Embedded
  Address address;

  @ElementCollection
  List<String> subjectList = new ArrayList<String>();

  //Question16-->oneToMany--unidirectional
  @OneToMany(mappedBy = "author")
  Set<Book> bookSet = new HashSet<Book>();

  public Author() {
  }
```

```
public Author(String firstName, String lastName, Integer age, Date dob) {
    this.firstName = firstName;
    this.lastName = lastName;
    this.age = age;
    this.dob = dob;
}

public Author(Integer id, String firstName, String lastName, Integer age) {
    this.id = id;
    this.firstName = firstName;
    this.lastName = lastName;
    this.age = age;
}

public Address getAddress() {
    return address;
}

public void setAddress(Address address) {
    this.address = address;
}

public Date getDob() {
    return dob;
}

public void setDob(Date dob) {
    this.dob = dob;
}

public Integer getId() {
    return id;
}

public void setId(Integer id) {
    this.id = id;
}

public String getFirstName() {
    return firstName;
}

public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public String getLastName() {
    return lastName;
}
```

```
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public Integer getAge() {
        return age;
    }

    public void setAge(Integer age) {
        this.age = age;
    }

    public List<String> getSubjectList() {
        return subjectList;
    }

    public void setSubjectList(List<String> subjectList) {
        this.subjectList = subjectList;
    }

    public Set<Book> getBookSet() {
        return bookSet;
    }

    public void setBookSet(Set<Book> bookSet) {
        this.bookSet = bookSet;
    }

    @Override
    public String toString() {
        return "Author{" +
                "id=" + id +
                ", firstName='" + firstName + '\'' +
                ", lastName='" + lastName + '\'' +
                ", age=" + age +
                ", dob=" + dob +
                ", address=" + address +
                ", subjectList=" + subjectList +
                ", bookSet=" + bookSet +
                '}';
    }
}
```

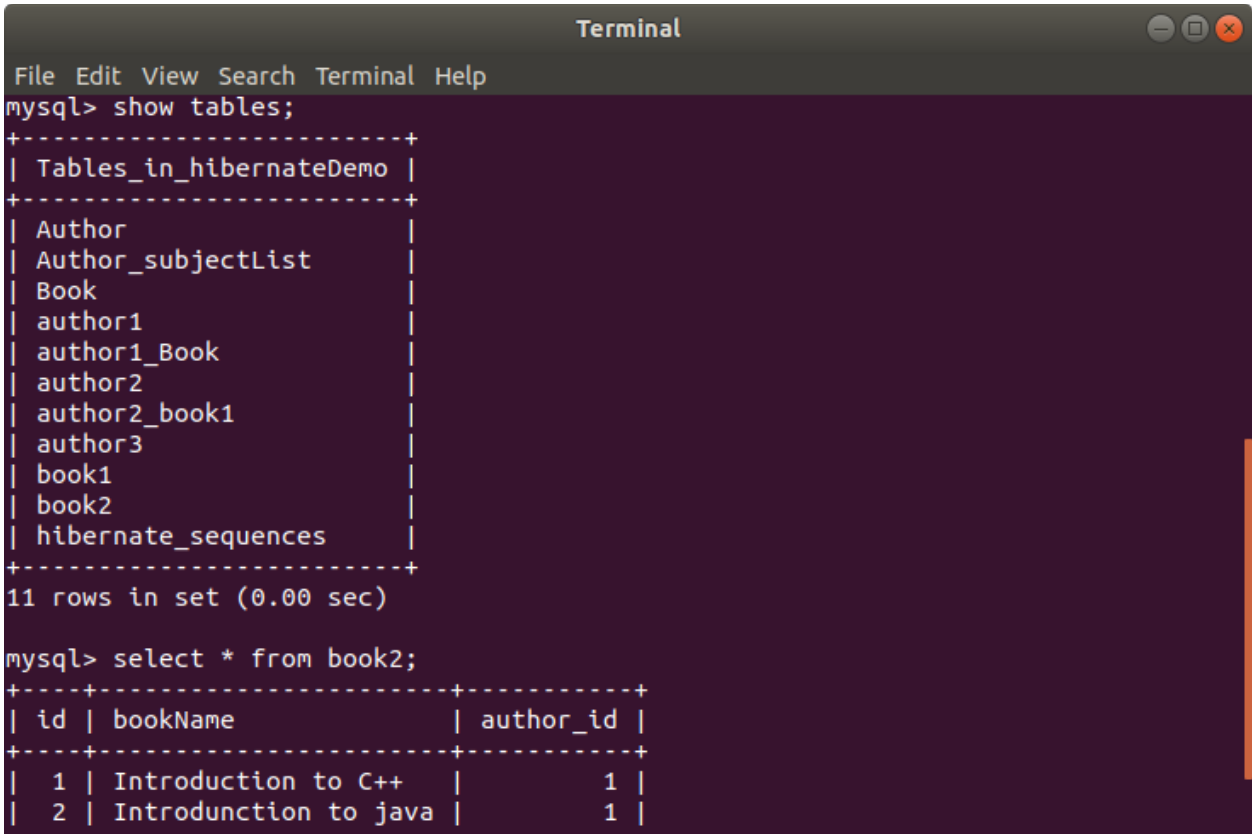
Book.java

Same as in previous case

Main.java

Same as in previous cas

Output


Terminal
File Edit View Search Terminal Help
mysql> show tables;
+-------------------------+
| Tables_in_hibernateDemo |
+-------------------------+
| Author |
| Author_subjectList |
| Book |
| author1 |
| author1_Book |
| author2 |
| author2_book1 |
| author3 |
| book1 |
| book2 |
| hibernate_sequences |
+-------------------------+
11 rows in set (0.00 sec)
mysql> select * from book2;
+----+-----------------------+-----------+
| id | bookName | author_id |
+----+-----------------------+-----------+
| 1 | Introduction to C++ | 1 |
| 2 | Introdunction to java | 1 |

Cascade
Hibernate.cfg.xml

```
<mapping class="com.demo.hibernate.question16.cascade.Book"/>
<mapping class="com.demo.hibernate.question16.cascade.Author"/>
```

Author.java

```
package com.demo.hibernate.question16.cascade;

import com.demo.hibernate.Address;


import javax.persistence.*;
import java.util.*;


@Entity
@Table(name = "author4")
public class Author {
  @Id
  @Column(name = "id")
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private Integer id;

  @Column(name = "first_name")
  private String firstName;

  @Column(name = "last_name")
  @Transient
  private String lastName;

  @Column(name = "age")
  private Integer age;

  @Column(name = "date_of_birth")
  @Temporal(TemporalType.DATE)
  private Date dob;


  @Embedded
  Address address;

  @ElementCollection
  List<String> subjectList = new ArrayList<String>();


  @OneToMany(mappedBy = "author",cascade = CascadeType.PERSIST)
  Set<Book> bookSet = new HashSet<Book>();
```

```
public Author() {
}

public Author(String firstName, String lastName, Integer age, Date dob) {
   this.firstName = firstName;
   this.lastName = lastName;
   this.age = age;
   this.dob = dob;
}

public Author(Integer id, String firstName, String lastName, Integer age) {
   this.id = id;
   this.firstName = firstName;
   this.lastName = lastName;
   this.age = age;
}

public Address getAddress() {
   return address;
}

public void setAddress(Address address) {
   this.address = address;
}

public Date getDob() {
   return dob;
}

public void setDob(Date dob) {
   this.dob = dob;
}

public Integer getId() {
   return id;
}

public void setId(Integer id) {
   this.id = id;
}

public String getFirstName() {
   return firstName;
}

public void setFirstName(String firstName) {
   this.firstName = firstName;
}

public String getLastName() {
```

```
    return lastName;
  }

  public void setLastName(String lastName) {
    this.lastName = lastName;
  }

  public Integer getAge() {
    return age;
  }

  public void setAge(Integer age) {
    this.age = age;
  }

  public List<String> getSubjectList() {
    return subjectList;
  }

  public void setSubjectList(List<String> subjectList) {
    this.subjectList = subjectList;
  }

  public Set<Book> getBookSet() {
    return bookSet;
  }

  public void setBookSet(Set<Book> bookSet) {
    this.bookSet = bookSet;
  }

  @Override
  public String toString() {
    return "Author{" +
        "id=" + id +
        ", firstName='" + firstName + '\'' +
        ", lastName='" + lastName + '\'' +
        ", age=" + age +
        ", dob=" + dob +
        ", address=" + address +
        ", subjectList=" + subjectList +
        ", bookSet=" + bookSet +
        '}';
  }
}
```

Book.java

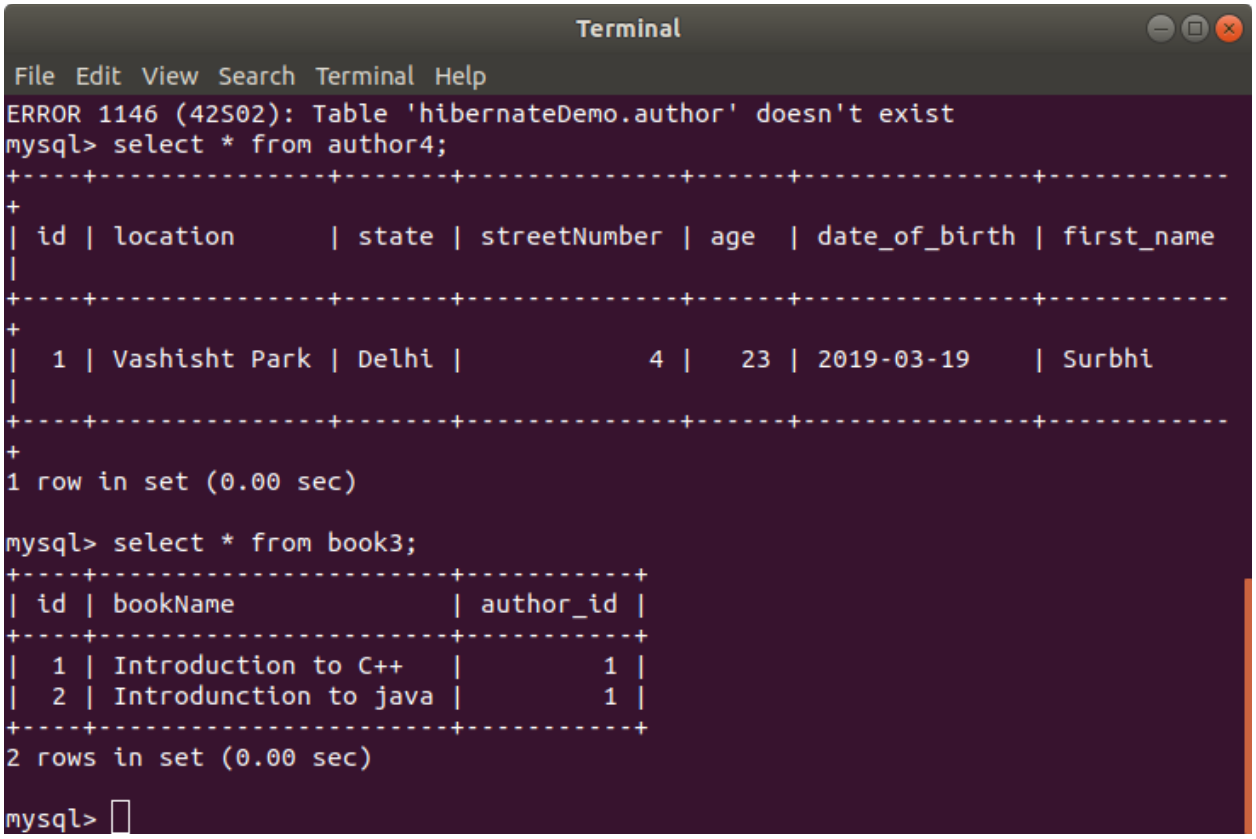
Same as in previous case

Main.java

```
package com.demo.hibernate.question16.cascade;
```

```java
import com.demo.hibernate.Address;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

import java.util.Arrays;
import java.util.Date;

public class Main {
  public static void main(String[] args) {

      Address address=new Address();
      address.setLocation("Vashisht Park");
      address.setState("Delhi");
      address.setStreetNumber(4);

      Author author=new Author("Surbhi","Garg",23,new Date());
      author.setAddress(address);
      author.setSubjectList(Arrays.asList("Science","Java","Maths"));

      Book book1=new Book();
      book1.setBookName("Introduction to C++");
      book1.setAuthor(author);


      Book book2=new Book();
      book2.setBookName("Introdunction to java");
      book2.setAuthor(author);

      author.getBookSet().addAll(Arrays.asList(book1,book2));


      SessionFactory sessionFactory=new Configuration().configure().buildSessionFactory();

      Session session=sessionFactory.openSession();
      session.beginTransaction();

      session.persist(author);


      session.getTransaction().commit();
      session.close();
      sessionFactory.close();
  }
}
```

Output


Terminal
File Edit View Search Terminal Help
ERROR 1146 (42S02): Table 'hibernateDemo.author' doesn't exist
mysql> select * from author4;
+----+---------------+-------+--------------+------+---------------+------------
+
| id | location | state | streetNumber | age | date_of_birth | first_name
|
+----+---------------+-------+--------------+------+---------------+------------
+
| 1 | Vashisht Park | Delhi | 4 | 23 | 2019-03-19 | Surbhi
|
+----+---------------+-------+--------------+------+---------------+------------
+
1 row in set (0.00 sec)
mysql> select * from book3;
+----+-----------------------+-----------+
| id | bookName | author_id |
+----+-----------------------+-----------+
| 1 | Introduction to C++ | 1 |
| 2 | Introdunction to java | 1 |
+----+-----------------------+-----------+
2 rows in set (0.00 sec)
mysql>

Q17. Implement Many to Many Mapping between Author and Book.

Solution

Hibernate.cfg.xml

```
<mapping class="com.demo.hibernate.question17.Book"/>
<mapping class="com.demo.hibernate.question17.Author"/>
```

Author.java

```
package com.demo.hibernate.question17;

import com.demo.hibernate.Address;

import javax.persistence.*;
import java.util.*;


@Entity
@Table(name = "author5")
public class Author {
  @Id
  @Column(name = "id")
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private Integer id;

  @Column(name = "first_name")
  private String firstName;

  @Column(name = "last_name")
  @Transient
  private String lastName;

  @Column(name = "age")
  private Integer age;

  @Column(name = "date_of_birth")
  @Temporal(TemporalType.DATE)
  private Date dob;


  @Embedded
  Address address;

  @ElementCollection
  List<String> subjectList = new ArrayList<String>();


  @ManyToMany(cascade = CascadeType.PERSIST)
  Set<Book> bookSet = new HashSet<Book>();

  public Author() {
  }
```

```
public Author(String firstName, String lastName, Integer age, Date dob) {
    this.firstName = firstName;
    this.lastName = lastName;
    this.age = age;
    this.dob = dob;
}

public Author(Integer id, String firstName, String lastName, Integer age) {
    this.id = id;
    this.firstName = firstName;
    this.lastName = lastName;
    this.age = age;
}

public Address getAddress() {
    return address;
}

public void setAddress(Address address) {
    this.address = address;
}

public Date getDob() {
    return dob;
}

public void setDob(Date dob) {
    this.dob = dob;
}

public Integer getId() {
    return id;
}

public void setId(Integer id) {
    this.id = id;
}

public String getFirstName() {
    return firstName;
}

public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public String getLastName() {
    return lastName;
}
```

```
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public Integer getAge() {
        return age;
    }

    public void setAge(Integer age) {
        this.age = age;
    }

    public List<String> getSubjectList() {
        return subjectList;
    }

    public void setSubjectList(List<String> subjectList) {
        this.subjectList = subjectList;
    }

    public Set<Book> getBookSet() {
        return bookSet;
    }

    public void setBookSet(Set<Book> bookSet) {
        this.bookSet = bookSet;
    }

    @Override
    public String toString() {
        return "Author{" +
                "id=" + id +
                ", firstName='" + firstName + '\'' +
                ", lastName='" + lastName + '\'' +
                ", age=" + age +
                ", dob=" + dob +
                ", address=" + address +
                ", subjectList=" + subjectList +
                ", bookSet=" + bookSet +
                '}';
    }
}
```

Book.java

```
package com.demo.hibernate.question17;

import javax.persistence.*;
import java.util.HashSet;
import java.util.Set;
```

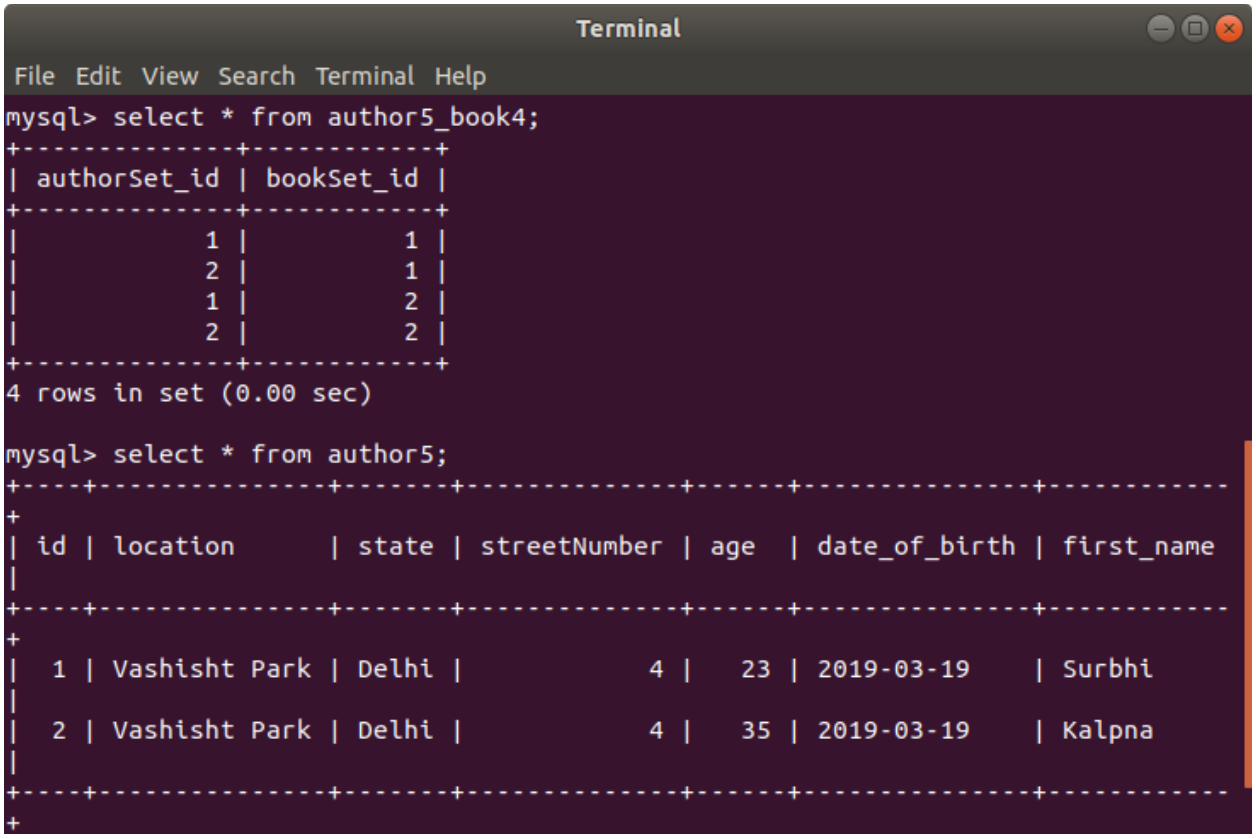
```java
@Entity
@Table(name = "book4")
public class Book {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private Integer id;
  private String bookName;
  @ManyToMany(mappedBy = "bookSet")
  private Set<Author> authorSet=new HashSet<Author>();

  public String getBookName() {
    return bookName;
  }

  public void setId(Integer id) {
    this.id = id;
  }

  public Set<Author> getAuthorSet() {
    return authorSet;
  }

  public void setAuthorSet(Set<Author> authorSet) {
    this.authorSet = authorSet;
  }

  public void setBookName(String bookName) {
    this.bookName = bookName;
  }

  public Integer getId() {
    return id;
  }

  @Override
  public String toString() {
    return "Book{" +
        "id=" + id +
        ", bookName='" + bookName + '\'' +
        '}';
  }
}
```

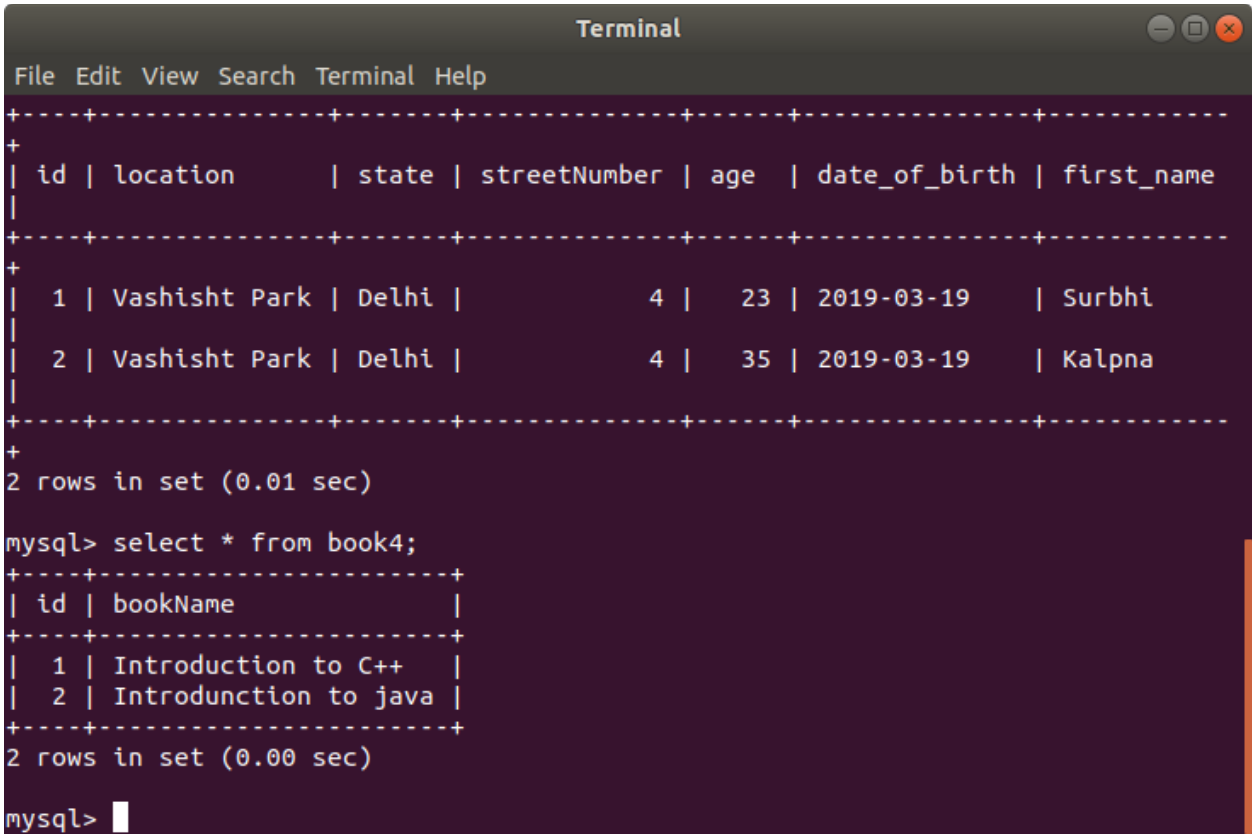
Main.java

```java
package com.demo.hibernate.question17;
import com.demo.hibernate.Address;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
```

```
import java.util.Arrays;
import java.util.Date;

public class Main {
  public static void main(String[] args) {

      Address address=new Address();
      address.setLocation("Vashisht Park");
      address.setState("Delhi");
      address.setStreetNumber(4);

      Author author=new Author("Surbhi","Garg",23,new Date());
      author.setAddress(address);
      author.setSubjectList(Arrays.asList("Science","Java","Maths"));


      Author author1=new Author("Kalpna","Sagar",35,new Date());
      author1.setAddress(address);
      author1.setSubjectList(Arrays.asList("Science","Java","Maths"));



      Book book1=new Book();
      book1.setBookName("Introduction to C++");
      book1.getAuthorSet().addAll(Arrays.asList(author,author1));


      Book book2=new Book();
      book2.setBookName("Introdunction to java");
      book2.getAuthorSet().addAll(Arrays.asList(author,author1));

      author.getBookSet().addAll(Arrays.asList(book1,book2));

      author1.getBookSet().addAll(Arrays.asList(book1,book2));


      SessionFactory sessionFactory=new Configuration().configure().buildSessionFactory();

      Session session=sessionFactory.openSession();
      session.beginTransaction();

      session.persist(author);
      session.persist(author1);


      session.getTransaction().commit();
      session.close();
      sessionFactory.close();
  }
}
```

Output


Terminal
File Edit View Search Terminal Help
mysql> select * from author5_book4;
+--------------+------------+
| authorSet_id | bookSet_id |
+--------------+------------+
1	1
2	1
1	2
2	2
+--------------+------------+
4 rows in set (0.00 sec)
mysql> select * from author5;
+----+---------------+-------+--------------+------+---------------+------------
+
| id | location | state | streetNumber | age | date_of_birth | first_name
|
+----+---------------+-------+--------------+------+---------------+------------
+
| 1 | Vashisht Park | Delhi | 4 | 23 | 2019-03-19 | Surbhi
|
| 2 | Vashisht Park | Delhi | 4 | 35 | 2019-03-19 | Kalpna
|
+----+---------------+-------+--------------+------+---------------+------------
+


Terminal
File Edit View Search Terminal Help
+----+---------------+-------+--------------+------+---------------+------------
+
| id | location | state | streetNumber | age | date_of_birth | first_name
|
+----+---------------+-------+--------------+------+---------------+------------
+
| 1 | Vashisht Park | Delhi | 4 | 23 | 2019-03-19 | Surbhi
|
| 2 | Vashisht Park | Delhi | 4 | 35 | 2019-03-19 | Kalpna
|
+----+---------------+-------+--------------+------+---------------+------------
+
2 rows in set (0.01 sec)
mysql> select * from book4;
+----+-----------------------+
| id | bookName |
+----+-----------------------+
| 1 | Introduction to C++ |
| 2 | Introdunction to java |
+----+-----------------------+
2 rows in set (0.00 sec)
mysql>