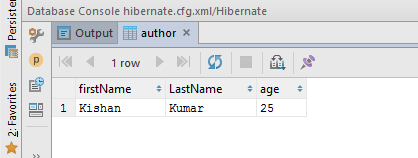
**Hibernate Exercise**

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

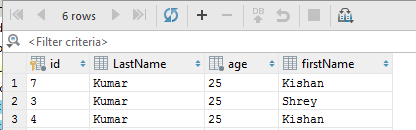
**import** javax.persistence.Entity;  
**import** javax.persistence.GeneratedValue;  
**import** javax.persistence.GenerationType;  
**import** javax.persistence.Id;  
  
@Entity  
**public class** Author {  
 @Id  
 @GeneratedValue (strategy = GenerationType.***AUTO***)  
 Integer **id**;  
 String **firstName**;  
 String **lastName**;  
 **int 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 int** getAge() {  
 **return age**;  
 }  
  
 **public void** setAge(**int** age) {  
 **this**.**age** = age;  
 }  
  
 @Override  
 **public** String toString() {  
 **return "Id: "**+**id**+**"\nFirst Name: "**+**firstName**+**"\nLast Name: "**+**lastName**+**"\nAge: "**+**age**;  
 }  
}

1. Perform CRUD operation for Author class.

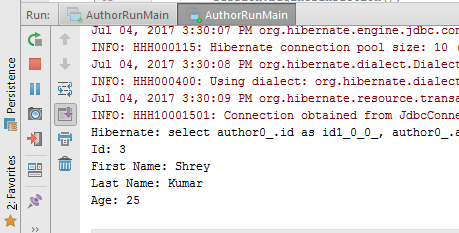
**import** org.hibernate.Session;  
**import** org.hibernate.SessionFactory;  
**import** org.hibernate.cfg.Configuration;  
**import** javax.persistence.Entity;  
  
**public class** AuthorMain {  
  
 **void** createQuery() {  
 SessionFactory sessionFactory = **new** org.hibernate.cfg.Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 Author author = **new** Author();  
 author.setFirstName(**"Kishan"**);  
 author.setLastName(**"Kumar"**);  
 author.setAge(25);  
 session.beginTransaction();  
 session.save(author);  
 session.getTransaction().commit();  
 session.close();  
  
 }  
  
}  
**class** AuthorRunMain{  
 **public static void** main(String[] args) {  
 AuthorMain authorMain = **new** AuthorMain();  
 authorMain.createQuery();  
  
 }  
}



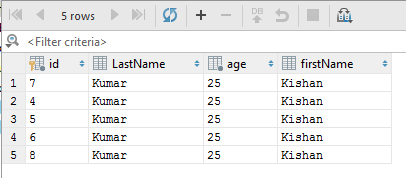
**void** updateQuery() {  
 SessionFactory sessionFactory = **new** Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 session.beginTransaction();  
 Author author = session.get(Author.**class**,3);  
 author.setFirstName(**"Shrey"**);  
 session.update(author);  
 session.getTransaction().commit();  
 session.close();  
  
}



**void** readQuery() {  
 SessionFactory sessionFactory = **new** Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 session.beginTransaction();  
 Author author = session.get(Author.**class**,3);  
 session.getTransaction().commit();  
 session.close();  
 System.***out***.println(author);  
  
 }



**void** deleteQuery() {  
 SessionFactory sessionFactory = **new** Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 session.beginTransaction();  
 Author author = session.get(Author.**class**,3);  
 session.delete(author);  
 session.getTransaction().commit();  
 session.close();  
 System.***out***.println(author);  
  
}



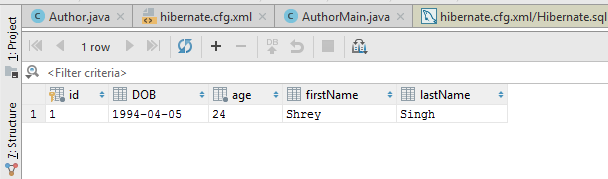
1. Use hbm2ddl create to introduce Date of Birth for Author.

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

**import** javax.persistence.\*;  
**import** java.util.Date;  
  
@Entity  
**public class** Author {  
 @Id  
 @GeneratedValue (strategy = GenerationType.***AUTO***)  
 Integer **id**;  
 String **firstName**;  
 String **lastName**;  
 @Temporal(TemporalType.***DATE***)  
 Date **DOB**;  
 **int 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** Date getDOB() {  
 **return DOB**;  
 }  
  
 **public void** setDOB(Date DOB) {  
 **this**.**DOB** = DOB;  
 }  
  
 **public int** getAge() {  
 **return age**;  
 }  
  
 **public void** setAge(**int** age) {  
 **this**.**age** = age;  
 }  
  
 @Override  
 **public** String toString() {  
 **return "Id: "**+**id**+**"\nFirst Name: "**+**firstName**+**"\nLast Name: "**+**lastName**+**"\nAge: "**+**age**;  
 }  
}

**void** createQuery(String fname,String lname,**int** age, java.util.Date dob) {  
 SessionFactory sessionFactory = **new** org.hibernate.cfg.Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 Author author = **new** Author();  
 author.setFirstName(fname);  
 author.setLastName(lname);  
 author.setAge(age);  
 author.setDOB(dob);  
 session.beginTransaction();  
 session.save(author);  
 session.getTransaction().commit();  
 session.close();  
  
 }

**class** AuthorRunMain{  
 **public static void** main(String[] args) {  
 AuthorMain authorMain = **new** AuthorMain();  
 java.util.Date date = **null**;  
 SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
 **try** {  
 date = simpleDateFormat.parse(**"1994-04-05"**);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
 authorMain.createQuery(**"Shrey"**,**"Singh"**,24,date);  
}

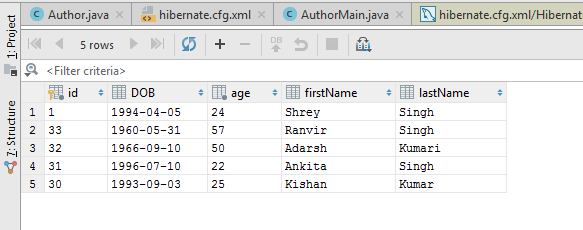


1. Use hbm2dll update to insert at least 4 records for Author.

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

**void** insertQuery(String fname, String lname, **int** age, java.util.Date date) {  
 SessionFactory sessionFactory = **new** org.hibernate.cfg.Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 Author author = **new** Author();  
 author.setFirstName(fname);  
 author.setLastName(lname);  
 author.setAge(age);  
 author.setDOB(date);  
 session.beginTransaction();  
 session.save(author);  
 session.getTransaction().commit();  
 session.close();  
  
}

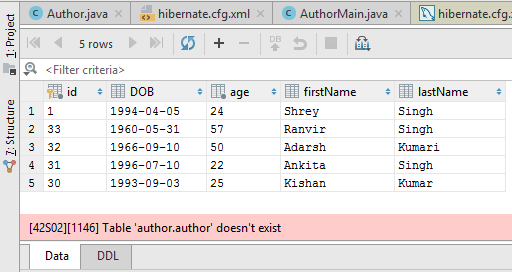
**class** AuthorRunMain {  
 **public static void** main(String[] args) {  
 AuthorMain authorMain = **new** AuthorMain();  
 String[] dob = {**"1993-09-03"**, **"1996-07-10"**, **"1966-09-10"**, **"1960-05-31"**};  
 String[] fname = {**"Kishan"**, **"Ankita"**, **"Adarsh"**, **"Ranvir"**};  
 String[] lname = {**"Kumar"**, **"Singh"**, **"Kumari"**, **"Singh"**};  
 **int**[] age = {25, 22, 50, 57};  
 java.util.Date date = **null**;  
 **for** (**int** i = 0; i < dob.**length**; i++) {  
   
 SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
 **try** {  
 date = simpleDateFormat.parse(dob[i]);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
authorMain.insertQuery(fname[i], lname[i], age[i], date);  
 }  
 }  
}

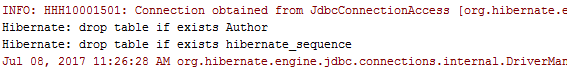


1. Perform hbm2dll create-drop by closing session factory.

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

**void** insertQuery(String fname, String lname, **int** age, java.util.Date date) {  
 SessionFactory sessionFactory = **new** org.hibernate.cfg.Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 Author author = **new** Author();  
 author.setFirstName(fname);  
 author.setLastName(lname);  
 author.setAge(age);  
 author.setDOB(date);  
 session.beginTransaction();  
 session.save(author);  
 session.getTransaction().commit();  
 session.close();  
 **sessionFactory.close();**  
}

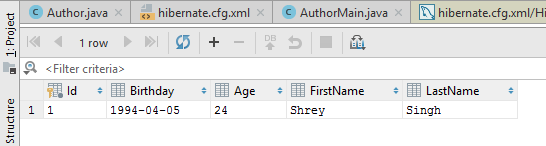




1. Rename all the fields using column annotation.

@Entity  
**public class** Author {  
 @Id  
 @GeneratedValue (strategy = GenerationType.***AUTO***)  
 @Column(name = **"Id"**)  
 Integer **id**;  
 @Column(name = **"FirstName"**)  
 String **firstName**;  
 @Column(name = **"LastName"**)  
 String **lastName**;  
 @Column(name = **"Age"**)  
 **int age**;  
 @Column(name = **"Birthday"**)  
 @Temporal(TemporalType.***DATE***)  
 Date **DOB**;

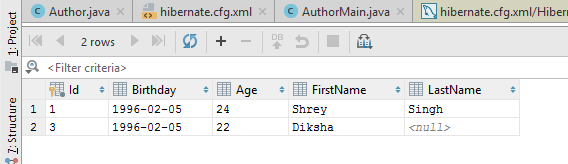
**class** AuthorRunMain {  
 **public static void** main(String[] args) {  
 AuthorMain authorMain = **new** AuthorMain();  
 java.util.Date date = **null**;  
 SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
 **try** {  
 date = simpleDateFormat.parse(**"1994-04-05"**);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
 authorMain.createQuery(**"Shrey"**,**"Singh"**,24,date);  
 }  
}



1. Mark lastName as @Transient.

@Entity  
**public class** Author {  
 @Id  
 @GeneratedValue (strategy = GenerationType.***AUTO***)  
 @Column(name = **"Id"**)  
 Integer **id**;  
 @Column(name = **"FirstName"**)  
 String **firstName**;  
 **@Transient**  
 @Column(name = **"LastName"**)  
 String **lastName**;  
 @Column(name = **"Age"**)  
 **int age**;  
 @Column(name = **"Birthday"**)  
 @Temporal(TemporalType.***DATE***)  
 Date **DOB**;

**class** AuthorRunMain {  
 **public static void** main(String[] args) {  
 AuthorMain authorMain = **new** AuthorMain();  
 java.util.Date date = **null**;  
 SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
 **try** {  
 date = simpleDateFormat.parse(**"1996-02-05"**);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
authorMain.createQuery(**"Diksha"**,**"Singh"**,22,date);  
 }  
}

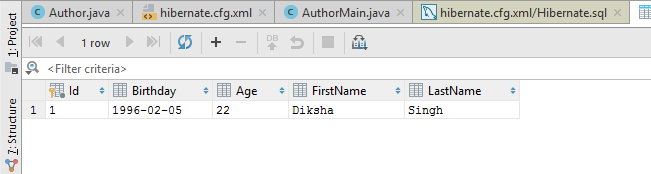


1. Use @Temporal for date of birth of Author.

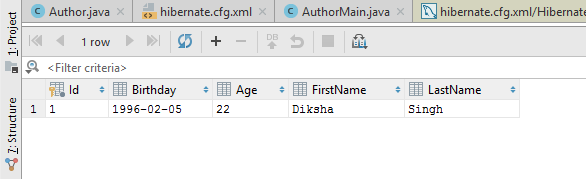
@Entity  
**public class** Author {  
 @Id  
 @GeneratedValue (strategy = GenerationType.***AUTO***)  
 @Column(name = **"Id"**)  
 Integer **id**;  
 @Column(name = **"FirstName"**)  
 String **firstName**;  
 @Transient  
 @Column(name = **"LastName"**)  
 String **lastName**;  
 @Column(name = **"Age"**)  
 **int age**;  
 @Column(name = **"Birthday"**)  
 **@Temporal(TemporalType.*DATE*)**  
 Date **DOB**;

1. Generate Id for Author Using IDENTITY and TABLE starategy.

@Entity  
**public class** Author {  
 @Id@GeneratedValue (strategy = GenerationType.***IDENTITY***)  
 @Column(name = **"Id"**)  
 Integer **id**;  
 @Column(name = **"FirstName"**)  
 String **firstName**;@Column(name = **"LastName"**)  
 String **lastName**;  
 @Column(name = **"Age"**)  
 **int age**;  
 @Column(name = **"Birthday"**)  
 @Temporal(TemporalType.***DATE***)  
 Date **DOB**;



@Entity  
**public class** Author {  
 @Id@GeneratedValue (strategy = GenerationType.***TABLE***)  
 @Column(name = **"Id"**)  
 Integer **id**;  
 @Column(name = **"FirstName"**)  
 String **firstName**;@Column(name = **"LastName"**)  
 String **lastName**;  
 @Column(name = **"Age"**)  
 **int age**;  
 @Column(name = **"Birthday"**)  
 @Temporal(TemporalType.***DATE***)  
 Date **DOB**;



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

@Entity  
**public class** Address {  
 @Id  
 @GeneratedValue(strategy = GenerationType.***AUTO***)  
 **int streetNo**;  
 String **location**;  
 String **state**;  
  
 **public int** getStreetNo() {  
 **return streetNo**;  
 }  
  
 **public void** setStreetNo(**int** streetNo) {  
 **this**.**streetNo** = streetNo;  
 }  
  
 **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;  
 }  
}

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

@Entity  
**public class** Author {  
 @Id@GeneratedValue (strategy = GenerationType.***TABLE***)  
 @Column(name = **"Id"**)  
 Integer **id**;  
 @Column(name = **"FirstName"**)  
 String **firstName**;@Column(name = **"LastName"**)  
 String **lastName**;  
 @Column(name = **"Age"**)  
 **int age**;  
 @Column(name = **"Birthday"**)  
 @Temporal(TemporalType.***DATE***)  
 Date **DOB**;  
 @Embedded  
 Address **address**;

@Entity  
@Embeddable  
**public class** Address {  
 @Id  
 @GeneratedValue(strategy = GenerationType.***AUTO***)  
 **int streetNo**;  
 String **location**;  
 String **state**;

**public int** getStreetNo() {  
 **return streetNo**;  
 }  
  
 **public void** setStreetNo(**int** streetNo) {  
 **this**.**streetNo** = streetNo;  
 }  
  
 **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;  
 }  
}

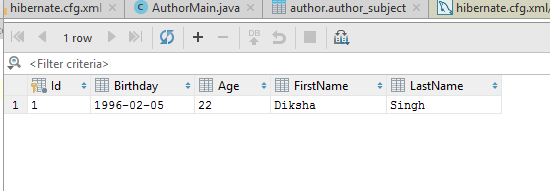
1. Introduce a List of subjects for author.

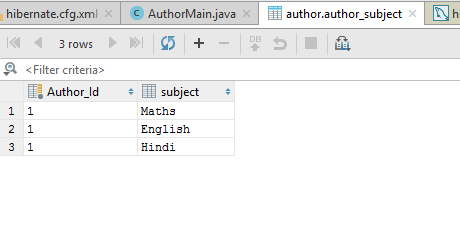
@Entity  
**public class** Author {  
 @Id@GeneratedValue (strategy = GenerationType.***TABLE***)  
 @Column(name = **"Id"**)  
 Integer **id**;  
 @Column(name = **"FirstName"**)  
 String **firstName**;@Column(name = **"LastName"**)  
 String **lastName**;  
 @Column(name = **"Age"**)  
 **int age**;  
 @Column(name = **"Birthday"**)  
 @Temporal(TemporalType.***DATE***)  
 Date **DOB**;@ElementCollection  
 List<String> **subject** = **new** ArrayList<String>();

1. Persist 3 subjects for each author.

**void** createQuery(String fname, String lname, **int** age, java.util.Date date,List<String> subject) {  
 SessionFactory sessionFactory = **new** org.hibernate.cfg.Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 Author author = **new** Author();  
 author.setFirstName(fname);  
 author.setLastName(lname);  
 author.setAge(age);  
 author.setDOB(date);  
 author.setSubject(subject);  
 session.beginTransaction();  
 session.save(author);  
 session.getTransaction().commit();  
 session.close();  
  
}

**class** AuthorRunMain {  
 **public static void** main(String[] args) {  
 AuthorMain authorMain = **new** AuthorMain();  
 java.util.Date date = **null**;  
 SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
 **try** {  
 date = simpleDateFormat.parse(**"1996-02-05"**);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
 List<String> subject = **new** ArrayList<>();  
 subject.add(**"Maths"**);  
 subject.add(**"English"**);  
 subject.add(**"Hindi"**);authorMain.createQuery(**"Diksha"**,**"Singh"**,22,date,subject);  
 }  
}





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

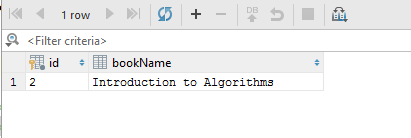
@Entity  
**public class** Book {  
 @Id  
 @GeneratedValue(strategy = GenerationType.***AUTO***)  
 **int id**;  
 String **bookName**;  
  
 **public int** getId() {  
 **return id**;  
 }  
  
 **public void** setId(**int** id) {  
 **this**.**id** = id;  
 }  
  
 **public** String getBookName() {  
 **return bookName**;  
 }  
  
 **public void** setBookName(String bookName) {  
 **this**.**bookName** = bookName;  
 }  
}

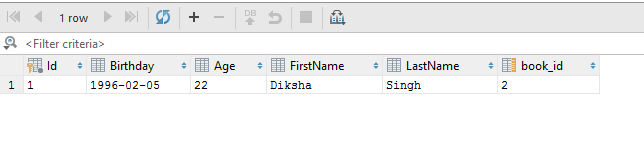
1. Implement One to One mapping between Author and Book.

@Entity  
**public class** Author {  
 @Id  
 @GeneratedValue (strategy = GenerationType.***AUTO***)  
@Column(name = **"Id"**)  
 Integer **id**;  
 @Column(name = **"FirstName"**)  
 String **firstName**;@Column(name = **"LastName"**)  
 String **lastName**;  
 @Column(name = **"Age"**)  
 **int age**;  
 @Column(name = **"Birthday"**)  
 @Temporal(TemporalType.***DATE***)  
 Date **DOB**;@ElementCollection  
 List<String> **subject** = **new** ArrayList<String>();  
  
 @OneToOne  
 Book **book**;  
  
 **public** Book getBook() {  
 **return book**;  
 }  
  
 **public void** setBook(Book book) {  
 **this**.**book** = book;  
 }

**void** createQuery(String fname, String lname, **int** age, java.util.Date date,List<String> subject,Book book) {  
 SessionFactory sessionFactory = **new** org.hibernate.cfg.Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 Author author = **new** Author();  
 author.setFirstName(fname);  
 author.setLastName(lname);  
 author.setAge(age);  
 author.setDOB(date);  
 author.setSubject(subject);  
 author.setBook(book);  
 session.beginTransaction();  
 session.save(author);  
 session.save(book);  
 session.getTransaction().commit();  
 session.close();  
  
}

**class** AuthorRunMain {  
 **public static void** main(String[] args) {  
 AuthorMain authorMain = **new** AuthorMain();  
   
 java.util.Date date = **null**;  
 SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
 **try** {  
 date = simpleDateFormat.parse(**"1996-02-05"**);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
 List<String> subject = **new** ArrayList<>();  
 subject.add(**"Maths"**);  
 subject.add(**"English"**);  
 subject.add(**"Hindi"**);  
 Book book = **new** Book();  
 book.setBookName(**"Introduction to Algorithms"**);authorMain.createQuery(**"Diksha"**,**"Singh"**,22,date,subject,book);  
 }  
}





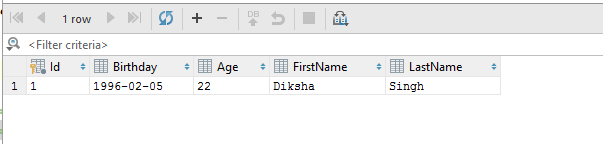
1. Implement One to Many Mapping between Author and Book(Unidirectional, BiDirectional and without additional table ) and  implement cascade save.

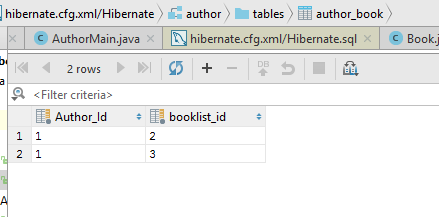
**UniDirectional**

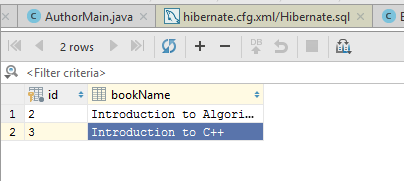
@Entity  
**public class** Author {  
 @Id  
 @GeneratedValue (strategy = GenerationType.***AUTO***)@Column(name = **"Id"**)  
 Integer **id**;  
 @Column(name = **"FirstName"**)  
 String **firstName**;  
@Column(name = **"LastName"**)  
 String **lastName**;  
 @Column(name = **"Age"**)  
 **int age**;  
 @Column(name = **"Birthday"**)  
 @Temporal(TemporalType.***DATE***)  
 Date **DOB**;  
@ElementCollection  
 List<String> **subject** = **new** ArrayList<String>();  
@OneToMany  
 List<Book> **booklist** = **new** ArrayList<>();  
  
 **public** List<Book> getBooklist() {  
 **return booklist**;  
 }  
  
 **public void** setBooklist(List<Book> booklist) {  
 **this**.**booklist** = booklist;  
 }

**void** createQuery(String fname, String lname, **int** age, java.util.Date date,List<String> subject) {  
 SessionFactory sessionFactory = **new** org.hibernate.cfg.Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 Author author = **new** Author();  
 author.setFirstName(fname);  
 author.setLastName(lname);  
 author.setAge(age);  
 author.setDOB(date);  
 author.setSubject(subject);  
 Book book1 = **new** Book();  
 book1.setBookName(**"Introduction to Algorithms"**);  
 Book book2 = **new** Book();  
author.getBooklist().add(book1);  
 author.getBooklist().add(book2);  
 session.beginTransaction();  
 session.save(author);  
 session.save(book1);  
 session.save(book2);  
 session.getTransaction().commit();  
 session.close();  
  
 }

**class** AuthorRunMain {  
 **public static void** main(String[] args) {  
 AuthorMain authorMain = **new** AuthorMain();  
  
 java.util.Date date = **null**;  
 SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
 **try** {  
 date = simpleDateFormat.parse(**"1996-02-05"**);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
 List<String> subject = **new** ArrayList<>();  
 subject.add(**"Maths"**);  
 subject.add(**"English"**);  
 subject.add(**"Hindi"**);  
authorMain.createQuery(**"Diksha"**,**"Singh"**,22,date,subject);  
 }  
}





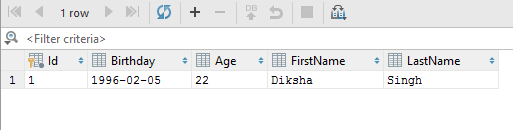


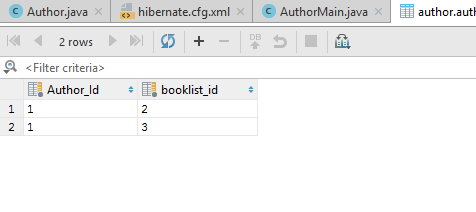
**BiDirectional**

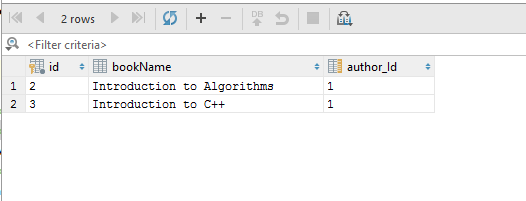
@Entity  
**public class** Author {  
 @Id  
 @GeneratedValue (strategy = GenerationType.***AUTO***)  
@Column(name = **"Id"**)  
 Integer **id**;  
 @Column(name = **"FirstName"**)  
 String **firstName**;  
@Column(name = **"LastName"**)  
 String **lastName**;  
 @Column(name = **"Age"**)  
 **int age**;  
 @Column(name = **"Birthday"**)  
 @Temporal(TemporalType.***DATE***)  
 Date **DOB**;@ElementCollection  
 List<String> **subject** = **new** ArrayList<String>();  
@OneToMany  
 List<Book> **booklist** = **new** ArrayList<>();  
  
 **public** List<Book> getBooklist() {  
 **return booklist**;  
 }  
  
 **public void** setBooklist(List<Book> booklist) {  
 **this**.**booklist** = booklist;  
 }

**void** createQuery(String fname, String lname, **int** age, java.util.Date date,List<String> subject) {  
 SessionFactory sessionFactory = **new** org.hibernate.cfg.Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 Author author = **new** Author();  
 author.setFirstName(fname);  
 author.setLastName(lname);  
 author.setAge(age);  
 author.setDOB(date);  
 author.setSubject(subject);  
 Book book1 = **new** Book();  
 book1.setBookName(**"Introduction to Algorithms"**);  
 book1.setAuthor(author);  
 Book book2 = **new** Book();  
 book2.setBookName(**"Introduction to C++"**);  
 book2.setAuthor(author);  
 author.getBooklist().add(book1);  
 author.getBooklist().add(book2);  
 session.beginTransaction();  
 session.save(author);  
 session.save(book1);  
 session.save(book2);  
 session.getTransaction().commit();  
 session.close();  
  
}

**class** AuthorRunMain {  
 **public static void** main(String[] args) {  
 AuthorMain authorMain = **new** AuthorMain();  
  
 java.util.Date date = **null**;  
 SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
 **try** {  
 date = simpleDateFormat.parse(**"1996-02-05"**);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
 List<String> subject = **new** ArrayList<>();  
 subject.add(**"Maths"**);  
 subject.add(**"English"**);  
 subject.add(**"Hindi"**);  
authorMain.createQuery(**"Diksha"**,**"Singh"**,22,date,subject);  
 }  
}





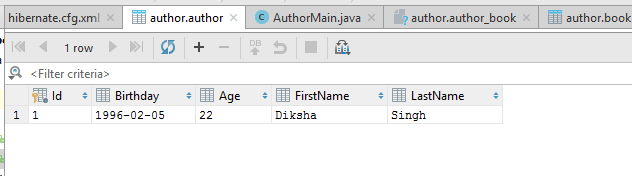


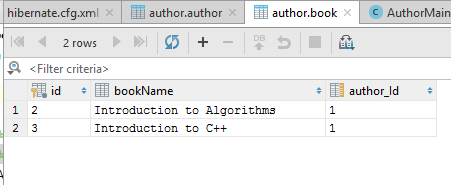
**Without Additional Table**

@Entity  
**public class** Author {  
 @Id  
 @GeneratedValue (strategy = GenerationType.***AUTO***)  
@Column(name = **"Id"**)  
 Integer **id**;  
 @Column(name = **"FirstName"**)  
 String **firstName**;  
@Column(name = **"LastName"**)  
 String **lastName**;  
 @Column(name = **"Age"**)  
 **int age**;  
 @Column(name = **"Birthday"**)  
 @Temporal(TemporalType.***DATE***)  
 Date **DOB**;  
@ElementCollection  
 List<String> **subject** = **new** ArrayList<String>();  
@OneToMany(mappedBy = **"author"**)  
 List<Book> **booklist** = **new** ArrayList<>();  
  
 **public** List<Book> getBooklist() {  
 **return booklist**;  
 }  
  
 **public void** setBooklist(List<Book> booklist) {  
 **this**.**booklist** = booklist;  
 }

**void** createQuery(String fname, String lname, **int** age, java.util.Date date,List<String> subject) {  
 SessionFactory sessionFactory = **new** org.hibernate.cfg.Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 Author author = **new** Author();  
 author.setFirstName(fname);  
 author.setLastName(lname);  
 author.setAge(age);  
 author.setDOB(date);  
 author.setSubject(subject);  
 Book book1 = **new** Book();  
 book1.setBookName(**"Introduction to Algorithms"**);  
 book1.setAuthor(author);  
 Book book2 = **new** Book();  
 book2.setBookName(**"Introduction to C++"**);  
 book2.setAuthor(author);  
 author.getBooklist().add(book1);  
 author.getBooklist().add(book2);  
 session.beginTransaction();  
 session.save(author);  
 session.save(book1);  
 session.save(book2);  
 session.getTransaction().commit();  
 session.close();  
  
}

**class** AuthorRunMain {  
 **public static void** main(String[] args) {  
 AuthorMain authorMain = **new** AuthorMain();  
  
 java.util.Date date = **null**;  
 SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
 **try** {  
 date = simpleDateFormat.parse(**"1996-02-05"**);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
 List<String> subject = **new** ArrayList<>();  
 subject.add(**"Maths"**);  
 subject.add(**"English"**);  
 subject.add(**"Hindi"**);  
authorMain.createQuery(**"Diksha"**,**"Singh"**,22,date,subject);  
 }  
}



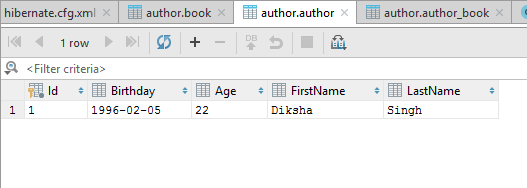


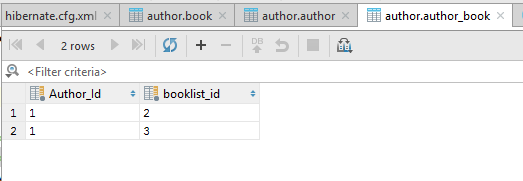
**Cascade**

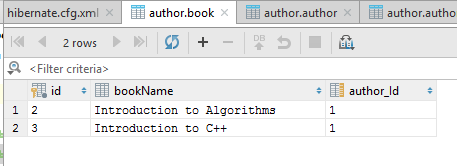
@Entity  
**public class** Author {  
 @Id  
 @GeneratedValue (strategy = GenerationType.***AUTO***)  
@Column(name = **"Id"**)  
 Integer **id**;  
 @Column(name = **"FirstName"**)  
 String **firstName**;@Column(name = **"LastName"**)  
 String **lastName**;  
 @Column(name = **"Age"**)  
 **int age**;  
 @Column(name = **"Birthday"**)  
 @Temporal(TemporalType.***DATE***)  
 Date **DOB**;  
@ElementCollection  
 List<String> **subject** = **new** ArrayList<String>();  
@OneToMany(cascade = CascadeType.***PERSIST***)  
 List<Book> **booklist** = **new** ArrayList<>();

**void** createQuery(String fname, String lname, **int** age, java.util.Date date,List<String> subject) {  
 SessionFactory sessionFactory = **new** org.hibernate.cfg.Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 Author author = **new** Author();  
 author.setFirstName(fname);  
 author.setLastName(lname);  
 author.setAge(age);  
 author.setDOB(date);  
 author.setSubject(subject);  
 Book book1 = **new** Book();  
 book1.setBookName(**"Introduction to Algorithms"**);  
 book1.setAuthor(author);  
 Book book2 = **new** Book();  
 book2.setBookName(**"Introduction to C++"**);  
 book2.setAuthor(author);  
 author.getBooklist().add(book1);  
 author.getBooklist().add(book2);  
 session.beginTransaction();  
 session.persist(author);  
session.getTransaction().commit();  
 session.close();  
  
 }

**class** AuthorRunMain {  
 **public static void** main(String[] args) {  
 AuthorMain authorMain = **new** AuthorMain();  
  
 java.util.Date date = **null**;  
 SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
 **try** {  
 date = simpleDateFormat.parse(**"1996-02-05"**);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
 List<String> subject = **new** ArrayList<>();  
 subject.add(**"Maths"**);  
 subject.add(**"English"**);  
 subject.add(**"Hindi"**);authorMain.createQuery(**"Diksha"**,**"Singh"**,22,date,subject);  
 }  
}







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

@Entity  
**public class** Author {  
 @Id  
 @GeneratedValue (strategy = GenerationType.***AUTO***)@Column(name = **"Id"**)  
 Integer **id**;  
 @Column(name = **"FirstName"**)  
 String **firstName**;  
@Column(name = **"LastName"**)  
 String **lastName**;  
 @Column(name = **"Age"**)  
 **int age**;  
 @Column(name = **"Birthday"**)  
 @Temporal(TemporalType.***DATE***)  
 Date **DOB**;  
@ElementCollection  
 List<String> **subject** = **new** ArrayList<String>();  
@ManyToMany(cascade = CascadeType.***PERSIST***)  
 List<Book> **booklist** = **new** ArrayList<>();

@Entity  
**public class** Book {  
 @Id  
 @GeneratedValue(strategy = GenerationType.***AUTO***)  
 **int id**;  
 String **bookName**;  
 */\*@ManyToOne  
Author author;  
\*/* @ManyToMany(mappedBy = **"booklist"**)  
 List<Author> **authorList** = **new** ArrayList<>();  
  
 **public** List<Author> getAuthorList() {  
 **return authorList**;  
 }  
  
 **public void** setAuthorList(List<Author> authorList) {  
 **this**.**authorList** = authorList;  
 }

**void** createQuery(String fname, String lname, **int** age, java.util.Date date,List<String> subject) {  
 SessionFactory sessionFactory = **new** org.hibernate.cfg.Configuration().configure().buildSessionFactory();  
 Session session = sessionFactory.openSession();  
 Author author = **new** Author();  
 author.setFirstName(fname);  
 author.setLastName(lname);  
 author.setAge(age);  
 author.setDOB(date);  
 author.setSubject(subject);  
 Book book1 = **new** Book();  
 book1.setBookName(**"Introduction to Algorithms"**);  
 Book book2 = **new** Book();  
 book2.setBookName(**"Introduction to C++"**);  
 author.getBooklist().add(book1);  
 author.getBooklist().add(book2);  
  
 Author author2 = **new** Author();  
 author2.setFirstName(**"Shrey"**);  
 author2.setLastName(**"Singh"**);  
 author2.setAge(23);  
 author2.setDOB(date);  
 author2.setSubject(subject);  
 author2.getBooklist().add(book1);  
 session.beginTransaction();  
 session.persist(author);  
 session.persist(author2);  
session.getTransaction().commit();  
 session.close();  
  
 }

**class** AuthorRunMain {  
 **public static void** main(String[] args) {  
 AuthorMain authorMain = **new** AuthorMain();  
  
 java.util.Date date = **null**;  
 SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**);  
 **try** {  
 date = simpleDateFormat.parse(**"1996-02-05"**);  
 } **catch** (ParseException e) {  
 e.printStackTrace();  
 }  
 List<String> subject = **new** ArrayList<>();  
 subject.add(**"Maths"**);  
 subject.add(**"English"**);  
 subject.add(**"Hindi"**);  
  
authorMain.createQuery(**"Diksha"**,**"Singh"**,22,date,subject);  
 }  
}

