Project name:BookPro

Book.java

```
package org.com;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table;
@Entity
@Table(name="Book_details")
public class Book {
		@Id
		@GeneratedValue
		@Column(name="book_id",nullable=false)
		private int id;
		@Column(name="book_name",nullable=false,length=100,unique=true)
		private String name;
		@Column(name="book_type",nullable=false,length=100)
		private String type;
		@Column(name="book_description")
		private String description;
		@Column(name="book_stud_id")
		private int stuId;
		@Column(name="book_author",nullable=false)
		private String author;
		@Column(name="book_price")
		private String price;
```

```
	public Book(int id, String name, String type, String description, int stuId, String author,
String price) {
		super();
		this.id = id;
		this.name = name;
		this.type = type;
		this.description = description;
		this.stuId = stuId;
		this.author = author;
		this.price = price;
	}
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public String getType() {
		return type;
	}
	public void setType(String type) {
		this.type = type;
```

```
	}
	public String getDescription() {
		return description;
	}
	public void setDescription(String description) {
		this.description = description;
	}
	public int getStuId() {
		return stuId;
	}
	public void setStuId(int stuId) {
		this.stuId = stuId;
	}
	public String getAuthor() {
		return author;
	}
	public void setAuthor(String author) {
		this.author = author;
	}
	public String getPrice() {
		return price;
	}
	public void setPrice(String price) {
		this.price = price;
	}
}
```

BookSolution.java

```
package org.com;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
public class Solution {
	public static void main(String[] args) throws IOException {
		BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
		SessionFactory sf = new Configuration().configure().buildSessionFactory();
		Session session = sf.openSession();
		int id=0, stuid, reader = 0;
		String name, type, desc, author, price;
		while (reader < 4) {
			session.beginTransaction();
			//System.out.println("Enter the book id");
			//id = Integer.valueOf(bf.readLine());
			System.out.println("Enter the book name");
			name = bf.readLine();
			System.out.println("Enter the book type");
			type = bf.readLine();
			System.out.println("Enter the book description");
			desc = bf.readLine();
			System.out.println("Enter the book student_id");
			stuid = Integer.valueOf(bf.readLine());
```

```
                System.out.println("Enter  the book author");
                author  = bf.readLine();
                System.out.println("Enter  the book price");
                price  = bf.readLine();
                Book book=new Book(id,  name,  type,  desc,  stuid,  author,price);
                session.save(book);
                session.getTransaction().commit();
                reader++;
            }
            session.close();
        }
}
```

Hibernate.cfg.xml

```
<?xml  version='1.0'  encoding='utf-8'?>
<!-- ~ Hibernate,  Relational  Persistence  for Idiomatic  Java ~ ~ License:
        GNU Lesser General  Public  License  (LGPL), version  2.1 or later. ~ See the
        lgpl.txt  file  in the root directory  or <http://www.gnu.org/licenses/lgpl-2.1.html>.  -->
<!DOCTYPE hibernate-configuration  PUBLIC
      "-//Hibernate/Hibernate  Configuration  DTD 3.0//EN"
      "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

        <session-factory>
```

```
        <!-- Database connection settings -->
        <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
        <property
name="connection.url">jdbc:mysql://localhost:3306/sample</property>
        <property name="connection.username">root</property>
        <property name="connection.password"></property>

        <!-- JDBC connection pool (use the built-in) -->
        <property name="connection.pool_size">10</property>

        <!-- SQL dialect -->
        <property name="dialect">org.hibernate.dialect.MySQL5Dialect</property>

        <!-- Disable the second-level cache -->
        <property
name="cache.provider_class">org.hibernate.cache.internal.NoCacheProvider</property>

        <!-- Echo all executed SQL to stdout -->
        <property name="show_sql">true</property>

        <!-- Drop and re-create the database schema on startup -->
        <property name="hbm2ddl.auto">create</property>

        <!-- Names the annotated entity class -->
        <mapping class="org.com.Book" />
    </session-factory>
</hibernate-configuration>
```