09\_Hibernate\_Assignment2

**package** com.cognizant.fsd.spring.config;

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**import** java.time.LocalDate;

**import** org.springframework.context.annotation.AnnotationConfigApplicationContext;

**import** org.springframework.context.support.AbstractApplicationContext;

**import** com.cognizant.fsd.spring.model.Book;

**import** com.cognizant.fsd.spring.model.Subject;

**import** com.cognizant.fsd.spring.service.BookService;

**import** com.cognizant.fsd.spring.service.SubjectService;

**public** **class** AppMain {

**private** BookService bookService;

**private** SubjectService subjectService;

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

**new** AppMain();

}

**public** AppMain() {

AbstractApplicationContext context = **new** AnnotationConfigApplicationContext(RootConfig.**class**);

**this**.bookService=(BookService) context.getBean("bookService");

**this**.subjectService=(SubjectService) context.getBean("subjectService");

**while** (**true**) {

System.***out***.println("\n");

System.***out***.println("enter '1' to add a new subject");

System.***out***.println("enter '2' to add a new book");

System.***out***.println("enter '3' to delete a subject");

System.***out***.println("enter '4' to delete a book");

System.***out***.println("enter '5' to search a subject");

System.***out***.println("enter '6' to search a book");

System.***out***.println("enter '7' to show all");

System.***out***.println("enter '8' to exit");

System.***out***.println("enter '10' to sort book by title");

System.***out***.println("enter '20' to sort subject by title");

System.***out***.println("enter '30' to sort book by publish date");

String input = gatValFromConsole("Enter your choise ");

select(Integer.*parseInt*(input));

}

}

**public** **void** select(**int** i) {

**switch**(i) {

**case** 1: subjectService.addSubject(addSubject());

**break**;

**case** 2: bookService.addBook(addBook());

**break**;

**case** 3: subjectService.deleteSubject(deleteSubject());

**break**;

**case** 4: bookService.deleteBook(deleteBook());

**break**;

**case** 5: Subject subject=subjectService.searchSubject(searchSubject());

System.***out***.println("Result :"+subject);

**break**;

**case** 6: Book book=bookService.searchBook(searchBook());

System.***out***.println("Result :"+book);

**break**;

**case** 7: showAll();

**break**;

**case** 10: bookService.fetchAllBookBySortingTitle().stream().forEach(System.***out***::println);

**break**;

**case** 20: subjectService.fetchAllSubjectBySortingTitle().stream().forEach(System.***out***::println);

**break**;

**case** 30: bookService.fetchAllBookBySortingDate().stream().forEach(System.***out***::println);

**break**;

**case** 8: exit();

**break**;

}

}

**private** **void** showAll() {

System.***out***.println("\n");

System.***out***.println("<------Total Subject List------->");

subjectService.fetchAllSubject().stream().forEach(System.***out***::println);

System.***out***.println("\n");

System.***out***.println("<------Total Book List------->");

bookService.fetchAllBook().stream().forEach(System.***out***::println);

}

**private** Subject addSubject() {

Subject subject = **new** Subject();;

subject.setSubjectId(Long.*parseLong*(gatValFromConsole("enter subject id")));

subject.setSubTitle(gatValFromConsole("enter subject title"));

subject.setDurationInHours(Integer.*parseInt*(gatValFromConsole("enter duration (In Hours)")));

System.***out***.println(subject.toString());

**return** subject;

}

**private** **long** deleteSubject() {

**long** subjectId=Long.*parseLong*(gatValFromConsole("enter the subject id to be deleted"));

**return** subjectId;

}

**private** **long** searchSubject() {

**long** subjectId=Long.*parseLong*(gatValFromConsole("enter the subject id for search"));

**return** subjectId;

}

**private** Book addBook() {

Book book = **new** Book();

book.setBookId(Long.*parseLong*(gatValFromConsole("enter book id")));

book.setTitle(gatValFromConsole("enter book title"));

//book.setPrice(Integer.parseInt(gatValFromConsole("enter book price")));

book.setVolume(Integer.*parseInt*(gatValFromConsole("enter book volume")));

book.setPublishDate(LocalDate.*parse*(gatValFromConsole("enter book publish date(yyyy-mm-dd) ")));

System.***out***.println(book.toString());

**return** book;

}

**private** **long** deleteBook() {

**long** bookId=Long.*parseLong*(gatValFromConsole("enter the book id to be deleted"));

**return** bookId;

}

**private** **long** searchBook() {

**long** bookId=Long.*parseLong*(gatValFromConsole("enter the book id for search "));

**return** bookId;

}

**private** String gatValFromConsole(String log) {

System.***out***.println(log);

String inputString = "";

**try** {

BufferedReader bufferRead = **new** BufferedReader(**new** InputStreamReader(System.***in***));

inputString = bufferRead.readLine();

} **catch** (IOException ex) {

ex.printStackTrace();

}

**return** inputString;

}

**private** **void** exit() {

System.*exit*(0);

}

}

package com.cognizant.fsd.spring.config;

import java.util.Properties;

import javax.sql.DataSource;

import org.apache.tomcat.dbcp.dbcp.BasicDataSource;

import org.hibernate.SessionFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.PropertySource;

import org.springframework.core.env.Environment;

import org.springframework.orm.hibernate4.HibernateTransactionManager;

import org.springframework.orm.hibernate4.LocalSessionFactoryBean;

import org.springframework.transaction.annotation.EnableTransactionManagement;

@Configuration

@EnableTransactionManagement

@PropertySource({ "classpath:mysql.properties" })

public class HibernateConfig {

@Autowired

private Environment env;

@Bean

public LocalSessionFactoryBean sessionFactory() {

LocalSessionFactoryBean sessionFactory = new LocalSessionFactoryBean();

sessionFactory.setDataSource(dataSource());

sessionFactory.setPackagesToScan(new String[] { "com.cognizant.fsd.spring.model" });

sessionFactory.setHibernateProperties(hibernateProperties());

return sessionFactory;

}

@Bean

public DataSource dataSource() {

BasicDataSource dataSource = new BasicDataSource();

dataSource.setDriverClassName(env.getProperty("jdbc.driverClassName"));

dataSource.setUrl(env.getProperty("jdbc.url"));

dataSource.setUsername(env.getProperty("jdbc.user"));

dataSource.setPassword(env.getProperty("jdbc.pass"));

return dataSource;

}

Properties hibernateProperties() {

return new Properties() {

{

setProperty("hibernate.hbm2ddl.auto",

env.getProperty("hibernate.hbm2ddl.auto"));

setProperty("hibernate.dialect",

env.getProperty("hibernate.dialect"));

setProperty("hibernate.globally\_quoted\_identifiers",

"true");

setProperty("hibernate.show\_sql",

"true");

}

};

}

@Bean

@Autowired

public HibernateTransactionManager transactionManager(SessionFactory s) {

HibernateTransactionManager txManager = new HibernateTransactionManager();

txManager.setSessionFactory(s);

return txManager;

}

}

**package** com.cognizant.fsd.spring.config;

**import** org.springframework.context.annotation.ComponentScan;

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.web.servlet.config.annotation.EnableWebMvc;

@Configuration

@ComponentScan(basePackages = { "com.cognizant.fsd.spring.config","com.cognizant.fsd.spring.controller","com.cognizant.fsd.spring.service","com.cognizant.fsd.spring.repository" })

**public** **class** RootConfig {

}

/\*\*

\*

\*/

package com.cognizant.fsd.spring.model;

import java.io.Serializable;

import java.time.LocalDate;

import java.util.HashSet;

import java.util.Set;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.ManyToMany;

import javax.persistence.Table;

import javax.persistence.Transient;

import org.springframework.format.annotation.DateTimeFormat;

/\*\*

\* @author Sahidul Hasan

\*

\*/

@Entity

@Table(name="book")

public class Book implements Serializable{

@Transient

private static final long serialVersionUID = 3667779253735136971L;

@Id

//@GeneratedValue(strategy=GenerationType.AUTO)

//private Long bid;

@Column(name="BOOKID")

private Long bookId;

@Column(name="TITLE")

private String title;

@Column(name="PRICE")

private Double price;

@Column(name="VOLUME")

private Integer volume;

@Column(name="PUBLISHDATE")

@DateTimeFormat(pattern = "yyyy-MM-dd")

private LocalDate publishDate;

@ManyToMany(mappedBy = "references",fetch=FetchType.EAGER)

private Set<Subject> references = new HashSet<>();

public Long getBookId() {

return bookId;

}

public void setBookId(Long bookId) {

this.bookId = bookId;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

public Integer getVolume() {

return volume;

}

public void setVolume(Integer volume) {

this.volume = volume;

}

public LocalDate getPublishDate() {

return publishDate;

}

public void setPublishDate(LocalDate publishDate) {

this.publishDate = publishDate;

}

public Set<Subject> getReferences() {

return references;

}

public void setReferences(Set<Subject> references) {

this.references = references;

}

@Override

public String toString() {

return "Book [bookId=" + bookId + ", title=" + title + ", price=" + price + ", volume=" + volume

+ ", publishDate=" + publishDate + "]";

}

}

package com.cognizant.fsd.spring.model;

import java.io.Serializable;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.ManyToMany;

import javax.persistence.Table;

import javax.persistence.Transient;

/\*\*

\* @author Sahidul Hasan

\*

\*/

@Entity

@Table(name="subject")

public class Subject implements Serializable{

@Transient

private static final long serialVersionId = 1L;

@Id

//@GeneratedValue(strategy=GenerationType.AUTO)

//private Long sid;

@Column(name="SUBJECTID")

private Long subjectId;

@Column(name="SUBTITLE")

private String subTitle;

@Column(name="DURATIONINHOURS")

private Integer durationInHours;

@ManyToMany(fetch=FetchType.EAGER)

private Set<Book> references=new HashSet<Book>();

@Transient

private List<Long> bookList;

public Long getSubjectId() {

return subjectId;

}

public void setSubjectId(Long subjectId) {

this.subjectId = subjectId;

}

public String getSubTitle() {

return subTitle;

}

public void setSubTitle(String subTitle) {

this.subTitle = subTitle;

}

public Integer getDurationInHours() {

return durationInHours;

}

public void setDurationInHours(Integer durationInHours) {

this.durationInHours = durationInHours;

}

public Set<Book> getReferences() {

return references;

}

public void setReferences(Set<Book> references) {

this.references = references;

}

@Override

public String toString() {

return "Subject [subjectId=" + subjectId + ", subTitle=" + subTitle + ", durationInHours=" + durationInHours

+ ", references=" + references + "]";

}

public List<Long> getBookList() {

return bookList;

}

public void setBookList(List<Long> bookList) {

this.bookList = bookList;

}

}

package com.cognizant.fsd.spring.service;

import java.util.List;

import com.cognizant.fsd.spring.model.Book;

public interface BookService {

public Book addBook(Book book);

public boolean deleteBook(long bookId);

public Book searchBook(long bookId);

public List<Book> fetchAllBook();

public List<Book> fetchAllBookBySortingDate();

public List<Book> fetchAllBookBySortingTitle();

}

package com.cognizant.fsd.spring.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.cognizant.fsd.spring.model.Book;

import com.cognizant.fsd.spring.repository.BookRepository;

@Service("bookService")

@Transactional

public class BookServiceImpl implements BookService {

@Autowired

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

@Override

public Book addBook(Book book) {

return bookRepository.addBook(book);

}

@Override

public boolean deleteBook(long bookId) {

return bookRepository.deleteBook(bookId);

}

@Override

public Book searchBook(long bookId) {

return bookRepository.searchBook(bookId);

}

@Override

public List<Book> fetchAllBook() {

return bookRepository.fetchAllBook();

}

@Override

public List<Book> fetchAllBookBySortingDate() {

return bookRepository.fetchAllBookBySortingDate();

}

@Override

public List<Book> fetchAllBookBySortingTitle() {

return bookRepository.fetchAllBookBySortingTitle();

}

}

package com.cognizant.fsd.spring.service;

import java.util.List;

import com.cognizant.fsd.spring.model.Subject;

public interface SubjectService {

public Subject addSubject(Subject subject);

public boolean deleteSubject(long subjectId);

public Subject searchSubject(long subjectId);

public List<Subject> fetchAllSubject();

public List<Subject> fetchAllSubjectBySortingTitle();

}

package com.cognizant.fsd.spring.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.cognizant.fsd.spring.model.Subject;

import com.cognizant.fsd.spring.repository.SubjectRepository;

@Service("subjectService")

@Transactional

public class SubjectServiceImpl implements SubjectService{

@Autowired

private SubjectRepository subjectRepository;

public void setSubjectRepository(SubjectRepository subjectRepository) {

this.subjectRepository = subjectRepository;

}

@Override

public Subject addSubject(Subject subject) {

return subjectRepository.addSubject(subject);

}

@Override

public boolean deleteSubject(long subjectId) {

return subjectRepository.deleteSubject(subjectId);

}

@Override

public Subject searchSubject(long subjectId) {

return subjectRepository.searchSubject(subjectId);

}

@Override

public List<Subject> fetchAllSubject() {

return subjectRepository.fetchAllSubject();

}

@Override

public List<Subject> fetchAllSubjectBySortingTitle() {

return subjectRepository.fetchAllSubjectBySortingTitle();

}

}

# jdbc

jdbc.driverClassName=com.mysql.jdbc.Driver

jdbc.url=jdbc:mysql://localhost:3306/hibernate\_assignment\_2?createDatabaseIfNotExist=true

jdbc.user=root

jdbc.pass=pass@word1

# hibernate

hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect

hibernate.show\_sql=true

hibernate.hbm2ddl.auto=update

package com.cognizant.fsd.spring.repository;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.springframework.beans.factory.annotation.Autowired;

public abstract class AbstractDao {

@Autowired

private SessionFactory sessionFactory;

protected Session getSession() {

return sessionFactory.getCurrentSession();

}

public void persist(Object entity) {

getSession().saveOrUpdate(entity);

}

public void delete(Object entity) {

getSession().delete(entity);

}

}

package com.cognizant.fsd.spring.repository;

import java.util.List;

import com.cognizant.fsd.spring.model.Book;

public interface BookRepository {

public Book addBook(Book book);

public boolean deleteBook(long bookId);

public Book searchBook(long bookId);

public List<Book> fetchAllBook();

public List<Book> fetchAllBookBySortingDate();

public List<Book> fetchAllBookBySortingTitle();

}

package com.cognizant.fsd.spring.repository;

import java.util.List;

import org.hibernate.Criteria;

import org.hibernate.criterion.Order;

import org.hibernate.criterion.Restrictions;

import org.springframework.stereotype.Repository;

import com.cognizant.fsd.spring.model.Book;

@Repository("bookRepository")

public class BookRepositoryImpl extends AbstractDao implements BookRepository {

@Override

public Book addBook(Book book) {

super.persist(book);

return book;

}

@Override

public boolean deleteBook(long bookId) {

boolean flag = false;

try {

Book book = (Book) getSession().get(Book.class, bookId);

super.delete(book);

flag = true;

} catch (Exception e) {

flag = false;

}

return flag;

}

@Override

public Book searchBook(long bookId) {

Criteria criteria = getSession().createCriteria(Book.class, "BK");

criteria.add(Restrictions.eq("BK.bookId", bookId));

Object book = (Object) criteria.uniqueResult();

return book != null ? (Book) book : null;

}

@Override

public List<Book> fetchAllBook() {

Criteria criteria = getSession().createCriteria(Book.class, "BK");

List<Book> bookList = criteria.list();

return bookList;

}

@Override

public List<Book> fetchAllBookBySortingDate() {

Criteria criteria = getSession().createCriteria(Book.class, "BK");

criteria.addOrder(Order.asc("publishDate"));

List<Book> bookList = criteria.list();

return bookList;

}

@Override

public List<Book> fetchAllBookBySortingTitle() {

Criteria criteria = getSession().createCriteria(Book.class, "BK");

criteria.addOrder(Order.asc("title"));

List<Book> bookList = criteria.list();

return bookList;

}

}

package com.cognizant.fsd.spring.repository;

import java.util.List;

import com.cognizant.fsd.spring.model.Subject;

public interface SubjectRepository {

public Subject addSubject(Subject subject);

public boolean deleteSubject(long subjectId);

public Subject searchSubject(long subjectId);

public List<Subject> fetchAllSubject();

public List<Subject> fetchAllSubjectBySortingTitle();

}

**package** com.cognizant.fsd.spring.repository;

**import** java.util.List;

**import** org.hibernate.Criteria;

**import** org.hibernate.criterion.Order;

**import** org.springframework.stereotype.Repository;

**import** com.cognizant.fsd.spring.model.Book;

**import** com.cognizant.fsd.spring.model.Subject;

@Repository("subjectRepository")

**public** **class** SubjectRepositoryImpl **extends** AbstractDao **implements** SubjectRepository{

@Override

**public** Subject addSubject(Subject subject) {

**super**.persist(subject);

**return** subject;

}

@Override

**public** **boolean** deleteSubject(**long** subjectId) {

**boolean** flag=**false**;

**try** {

Subject subject=(Subject) getSession().load(Subject.**class**, subjectId);

**super**.delete(subject);

flag=**true**;

}**catch**(Exception e) {

flag=**false**;

}

**return** flag;

}

@Override

**public** Subject searchSubject(**long** subjectId) {

Object subject=getSession().get(Subject.**class**, subjectId);

**return** subject!=**null**?(Subject)subject:**null**;

}

@Override

**public** List<Subject> fetchAllSubject() {

Criteria criteria= getSession().createCriteria(Subject.**class**,"SUB");

List<Subject> subjectList=criteria.list();

**return** subjectList;

}

@Override

**public** List<Subject> fetchAllSubjectBySortingTitle() {

Criteria criteria = getSession().createCriteria(Subject.**class**, "SUB");

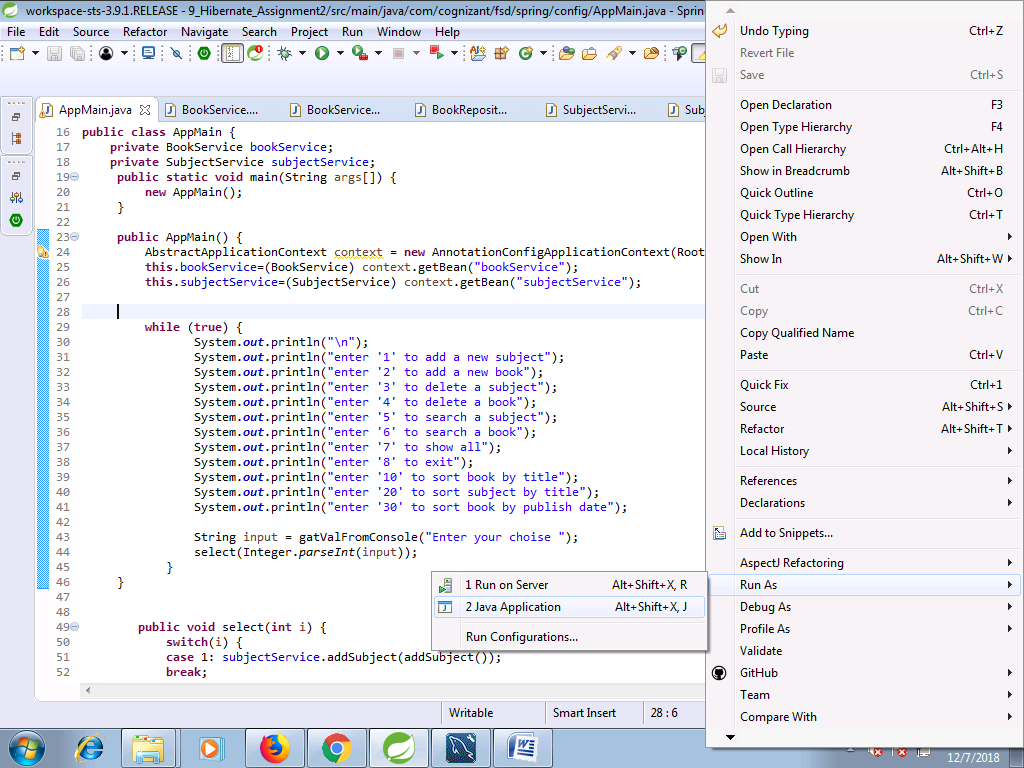
criteria.addOrder(Order.*asc*("subTitle"));

List<Subject> subjectList = criteria.list();

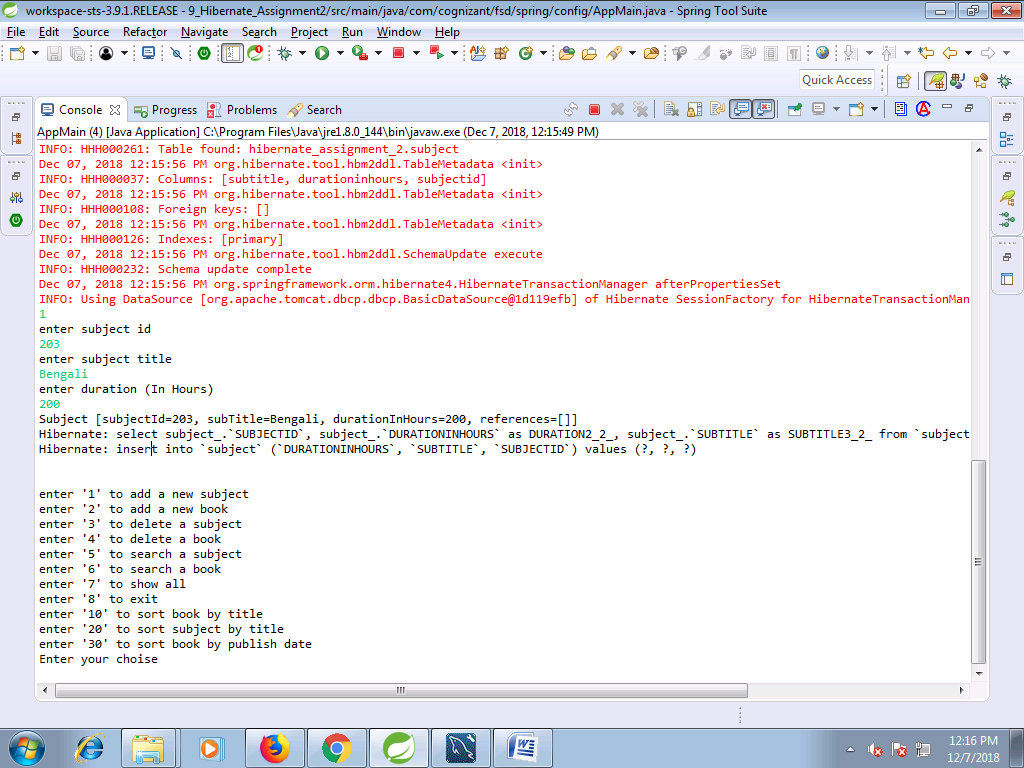
**return** subjectList;

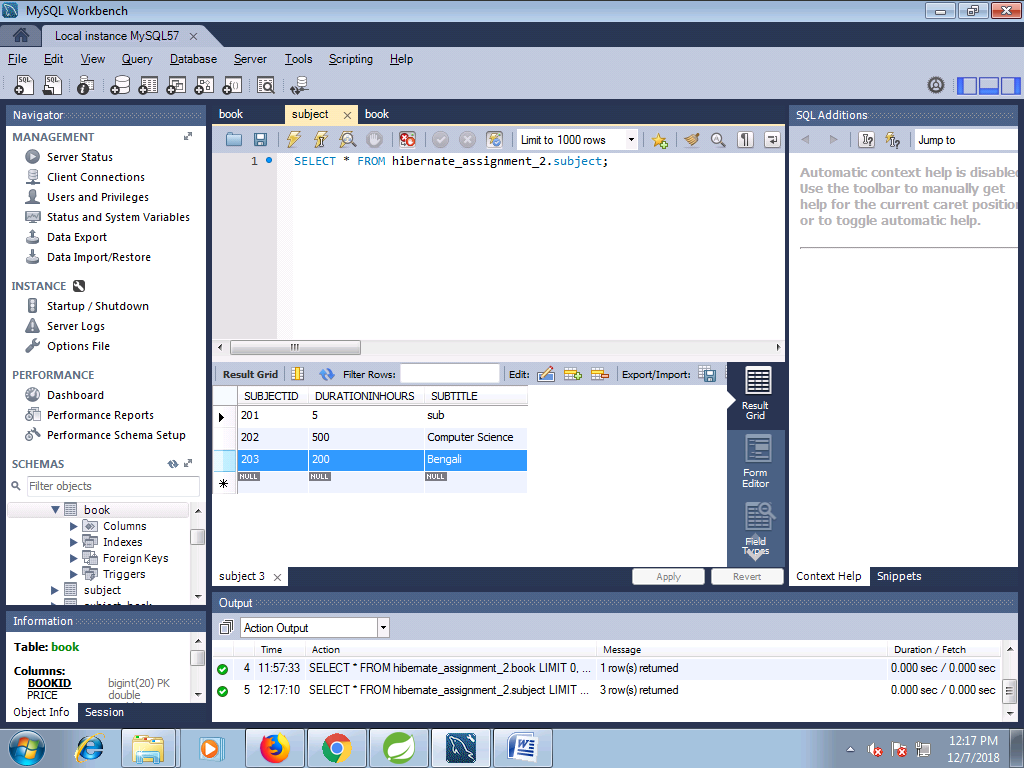
}

}

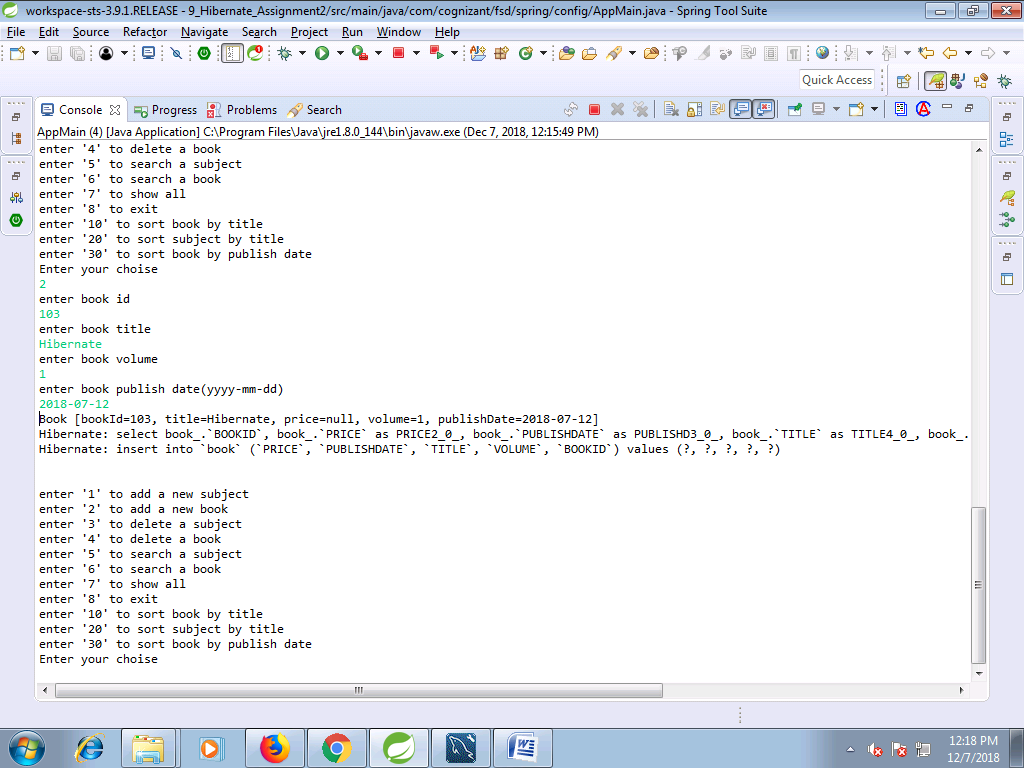


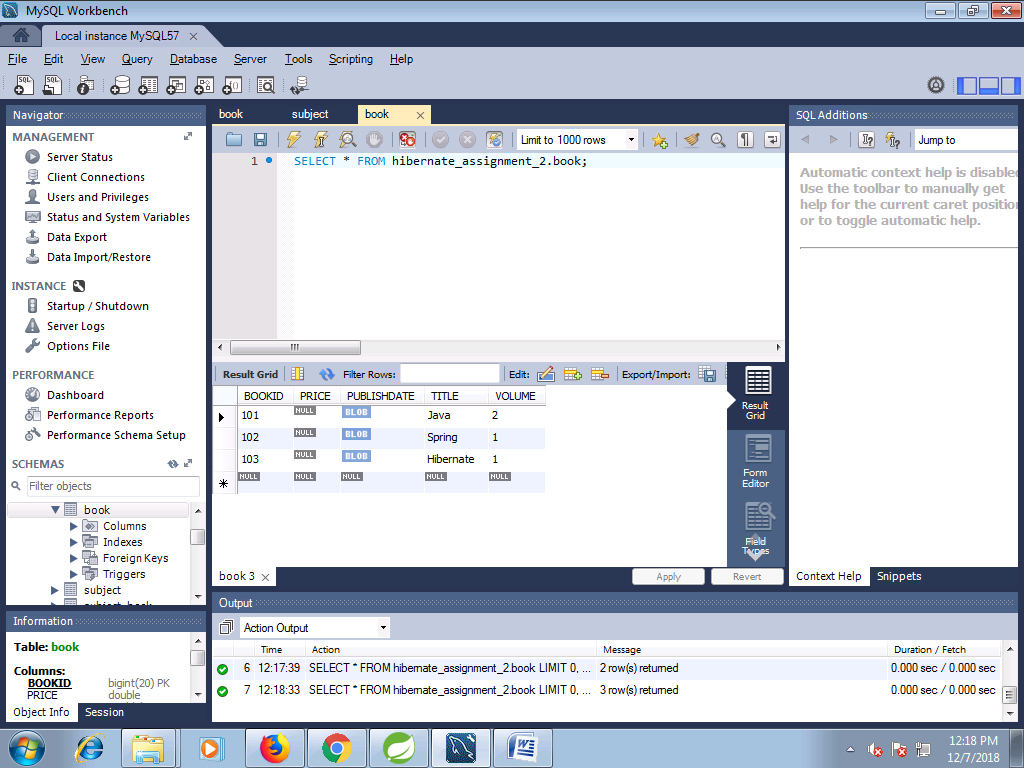
add a new subject



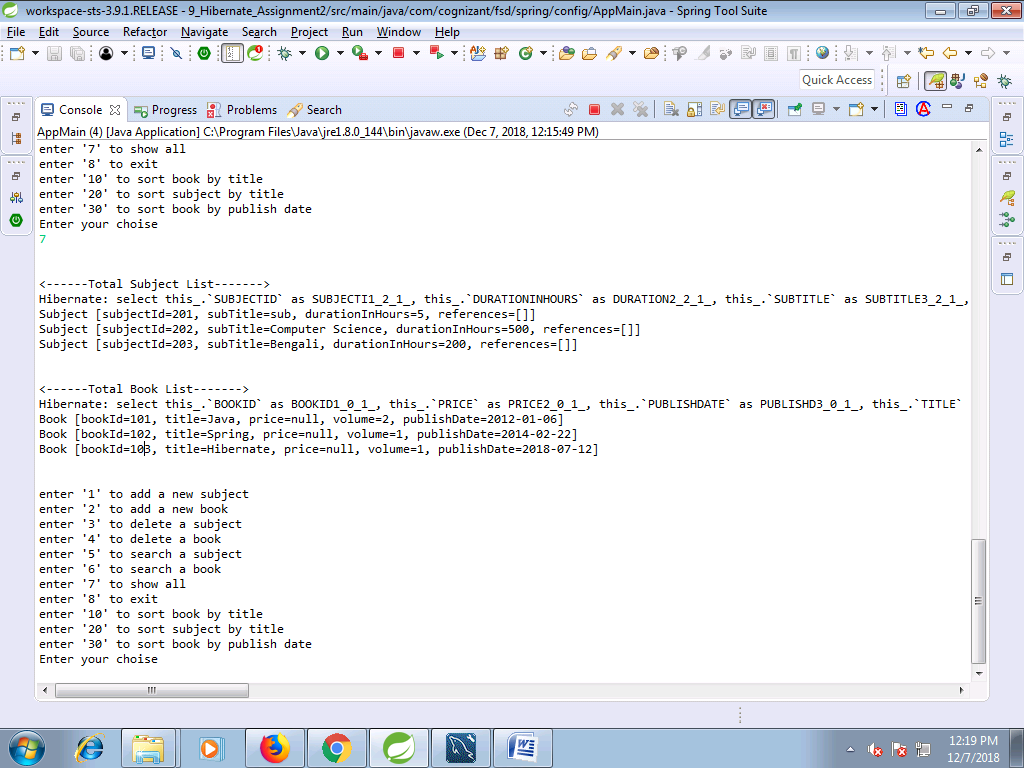


add new book

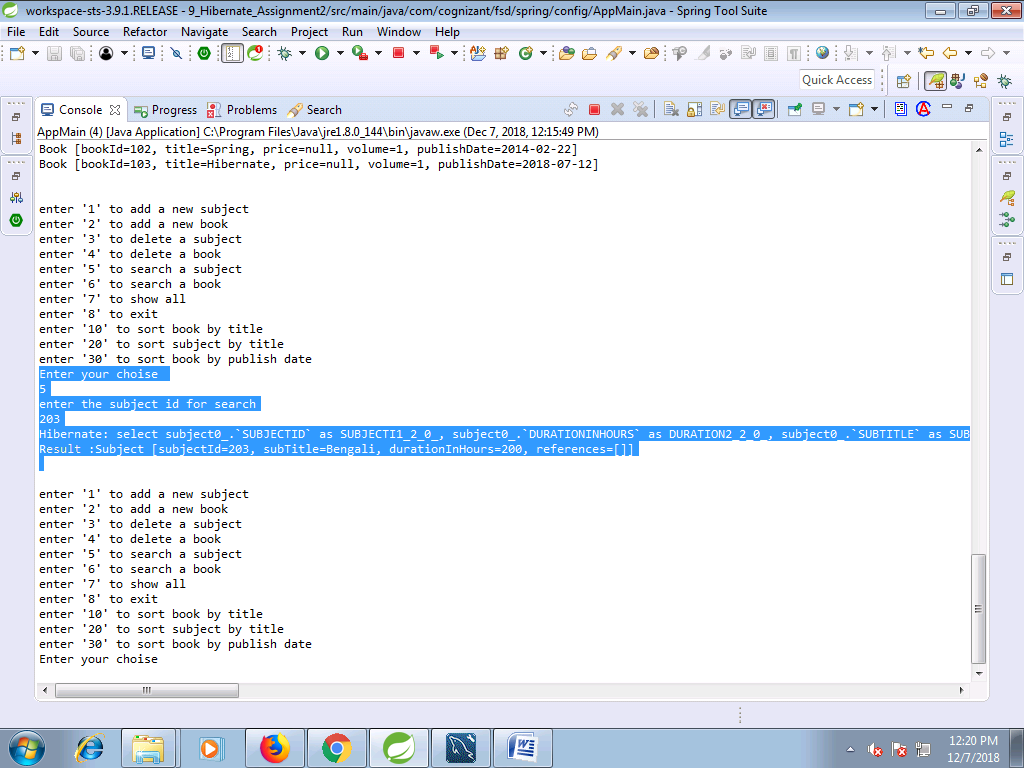




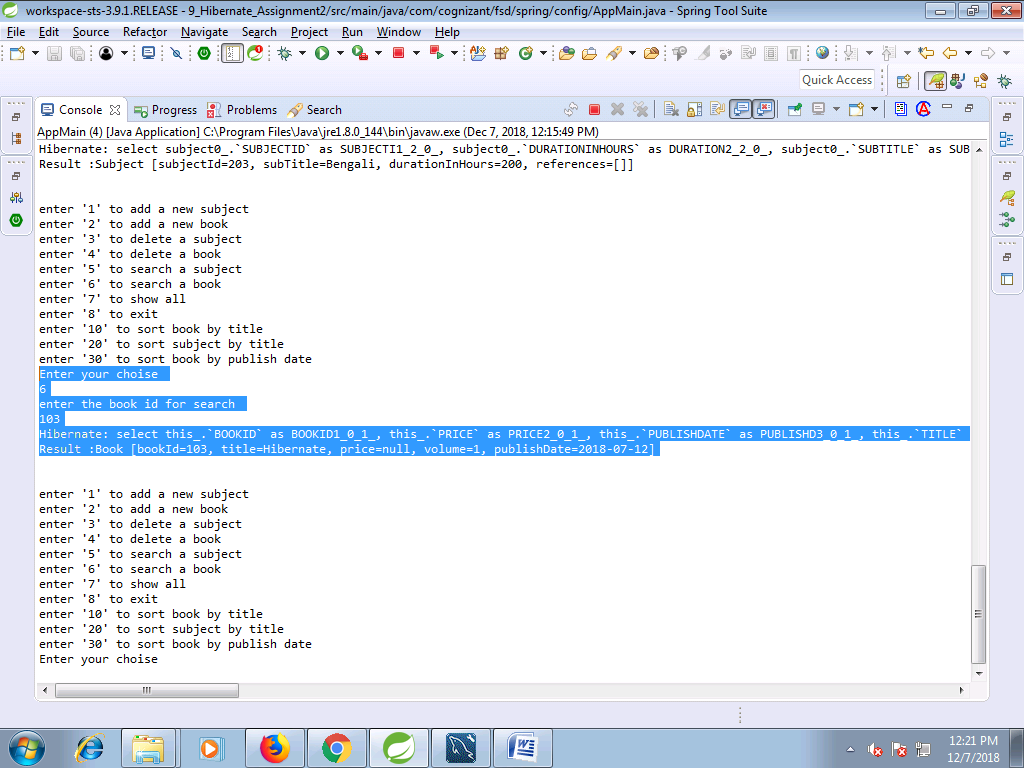
show all



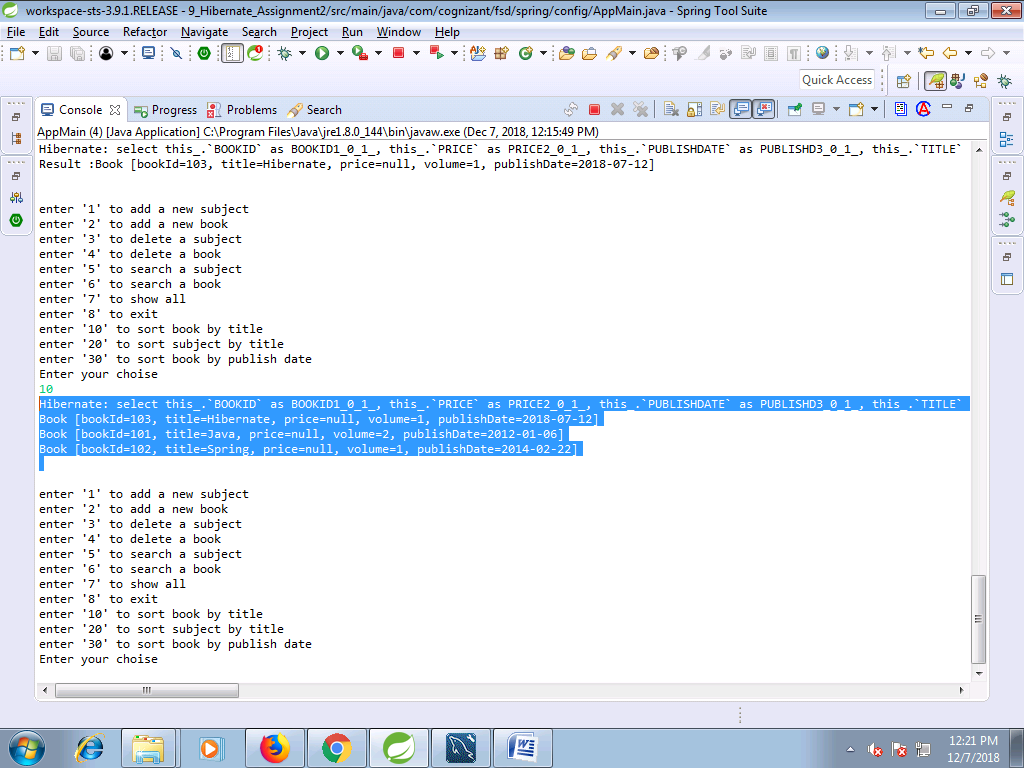
search a subject



search a book



sort book by title



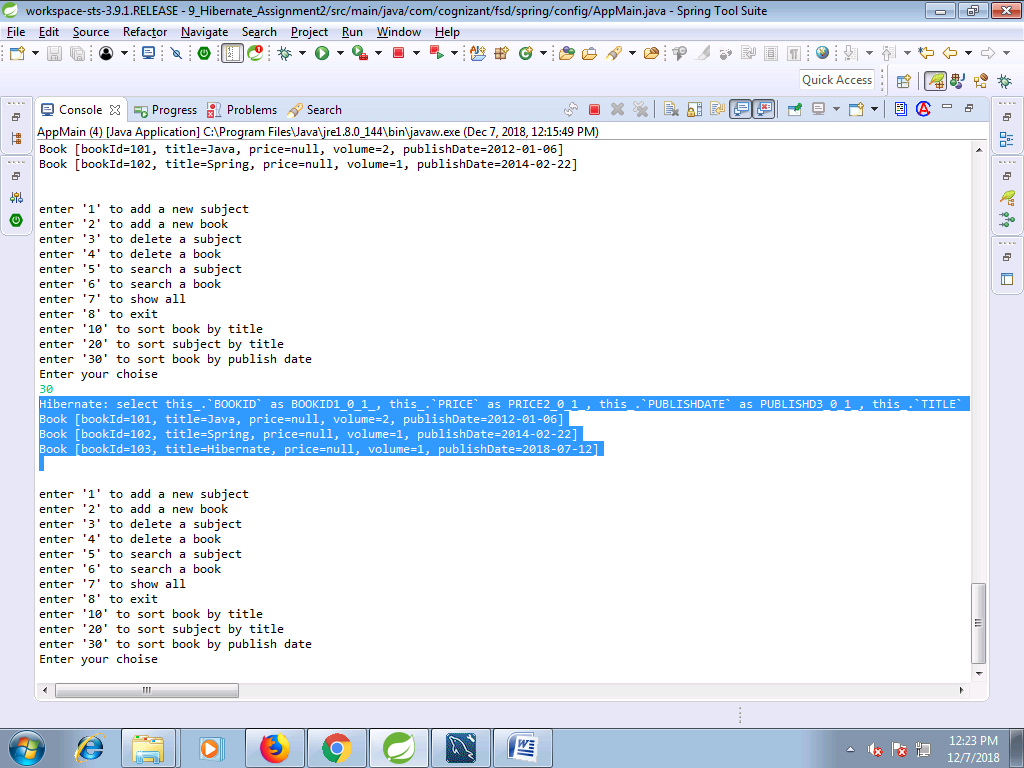
Hibernate: select this\_.`BOOKID` as BOOKID1\_0\_1\_, this\_.`PRICE` as PRICE2\_0\_1\_, this\_.`PUBLISHDATE` as PUBLISHD3\_0\_1\_, this\_.`TITLE` as TITLE4\_0\_1\_, this\_.`VOLUME` as VOLUME5\_0\_1\_, references2\_.`references\_BOOKID` as referenc2\_0\_3\_, subject3\_.`SUBJECTID` as referenc1\_1\_3\_, subject3\_.`SUBJECTID` as SUBJECTI1\_2\_0\_, subject3\_.`DURATIONINHOURS` as DURATION2\_2\_0\_, subject3\_.`SUBTITLE` as SUBTITLE3\_2\_0\_ from `book` this\_ left outer join `subject\_book` references2\_ on this\_.`BOOKID`=references2\_.`references\_BOOKID` left outer join `subject` subject3\_ on references2\_.`references\_SUBJECTID`=subject3\_.`SUBJECTID` order by this\_.`TITLE` asc

Book [bookId=103, title=Hibernate, price=null, volume=1, publishDate=2018-07-12]

Book [bookId=101, title=Java, price=null, volume=2, publishDate=2012-01-06]

Book [bookId=102, title=Spring, price=null, volume=1, publishDate=2014-02-22]

sort book by publish date



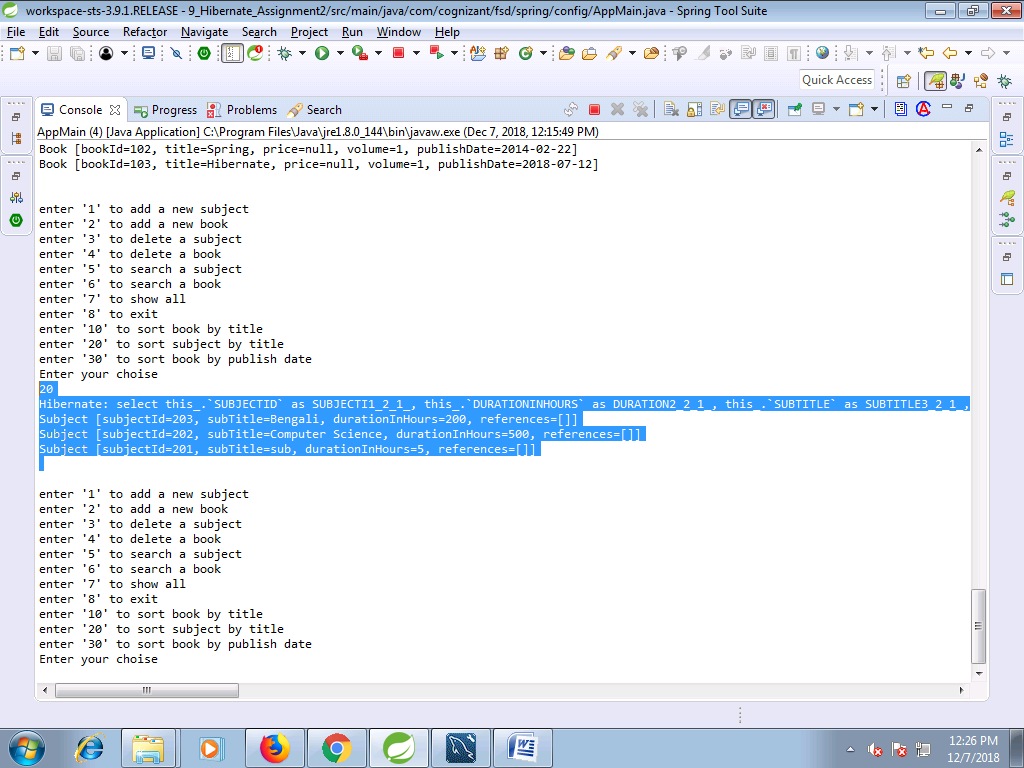
Hibernate: select this\_.`BOOKID` as BOOKID1\_0\_1\_, this\_.`PRICE` as PRICE2\_0\_1\_, this\_.`PUBLISHDATE` as PUBLISHD3\_0\_1\_, this\_.`TITLE` as TITLE4\_0\_1\_, this\_.`VOLUME` as VOLUME5\_0\_1\_, references2\_.`references\_BOOKID` as referenc2\_0\_3\_, subject3\_.`SUBJECTID` as referenc1\_1\_3\_, subject3\_.`SUBJECTID` as SUBJECTI1\_2\_0\_, subject3\_.`DURATIONINHOURS` as DURATION2\_2\_0\_, subject3\_.`SUBTITLE` as SUBTITLE3\_2\_0\_ from `book` this\_ left outer join `subject\_book` references2\_ on this\_.`BOOKID`=references2\_.`references\_BOOKID` left outer join `subject` subject3\_ on references2\_.`references\_SUBJECTID`=subject3\_.`SUBJECTID` order by this\_.`PUBLISHDATE` asc

Book [bookId=101, title=Java, price=null, volume=2, publishDate=2012-01-06]

Book [bookId=102, title=Spring, price=null, volume=1, publishDate=2014-02-22]

Book [bookId=103, title=Hibernate, price=null, volume=1, publishDate=2018-07-12]

sort subject by title



Hibernate: select this\_.`SUBJECTID` as SUBJECTI1\_2\_1\_, this\_.`DURATIONINHOURS` as DURATION2\_2\_1\_, this\_.`SUBTITLE` as SUBTITLE3\_2\_1\_, references2\_.`references\_SUBJECTID` as referenc1\_2\_3\_, book3\_.`BOOKID` as referenc2\_1\_3\_, book3\_.`BOOKID` as BOOKID1\_0\_0\_, book3\_.`PRICE` as PRICE2\_0\_0\_, book3\_.`PUBLISHDATE` as PUBLISHD3\_0\_0\_, book3\_.`TITLE` as TITLE4\_0\_0\_, book3\_.`VOLUME` as VOLUME5\_0\_0\_ from `subject` this\_ left outer join `subject\_book` references2\_ on this\_.`SUBJECTID`=references2\_.`references\_SUBJECTID` left outer join `book` book3\_ on references2\_.`references\_BOOKID`=book3\_.`BOOKID` order by this\_.`SUBTITLE` asc

Subject [subjectId=203, subTitle=Bengali, durationInHours=200, references=[]]

Subject [subjectId=202, subTitle=Computer Science, durationInHours=500, references=[]]

Subject [subjectId=201, subTitle=sub, durationInHours=5, references=[]]