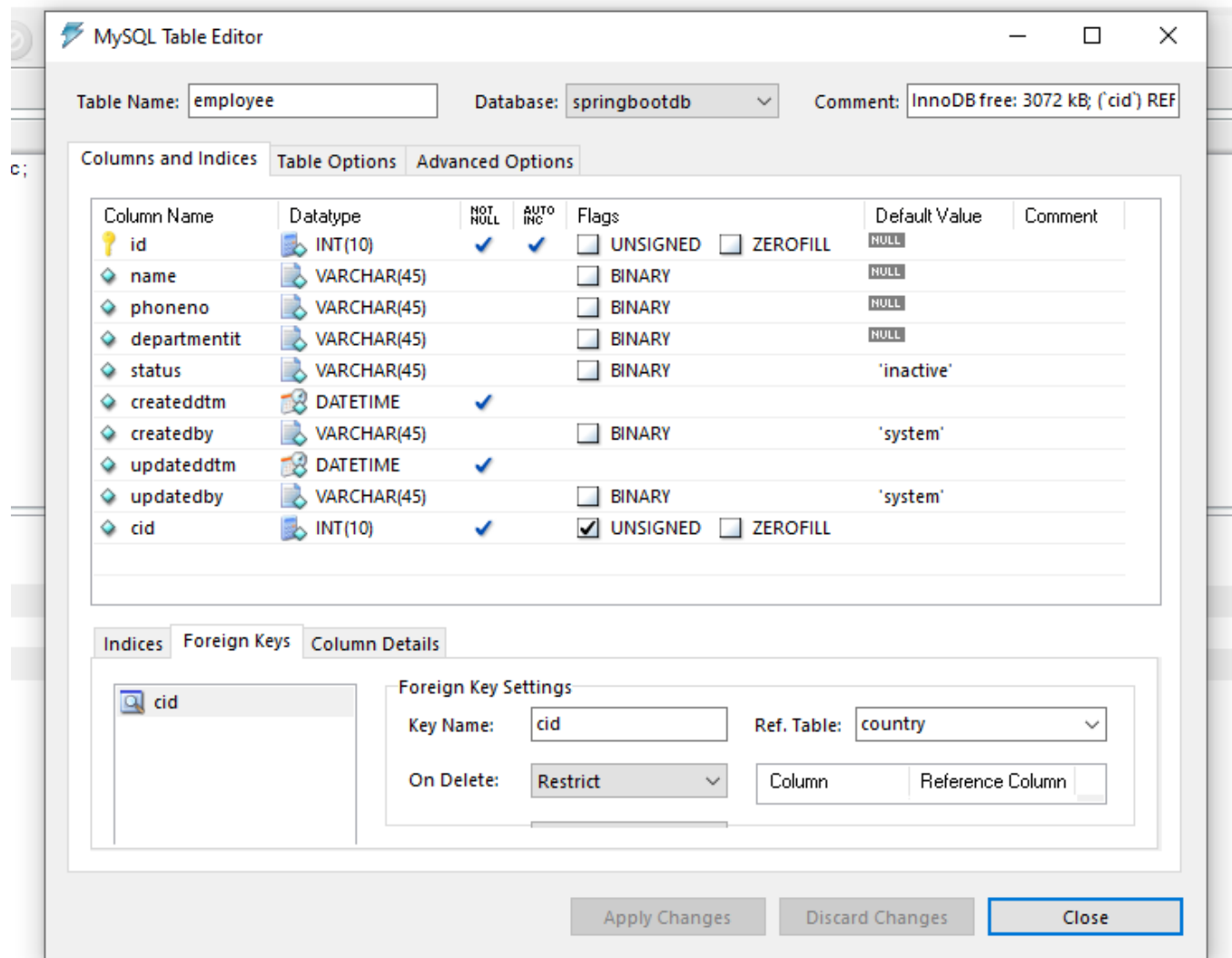


Assignment: Hibernate Lazy and Eager Load in Foreign Key

Table Structure:- Employee











est



MySQL Table Editor

Table Name: Database: Comment:

Columns and Indices | Table Options | Advanced Options

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
 id	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL	
 name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	NULL	
 phoneno	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	NULL	
 departmentit	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	NULL	
 status	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	'inactive'	
 createddtm	DATETIME	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
 createdby	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	'system'	
 updateddtm	DATETIME	<input checked="" type="checkbox"/>	<input type="checkbox"/>			
 updatedby	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY	'system'	
 cid	INT(10)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL		

Indices | Foreign Keys | Column Details

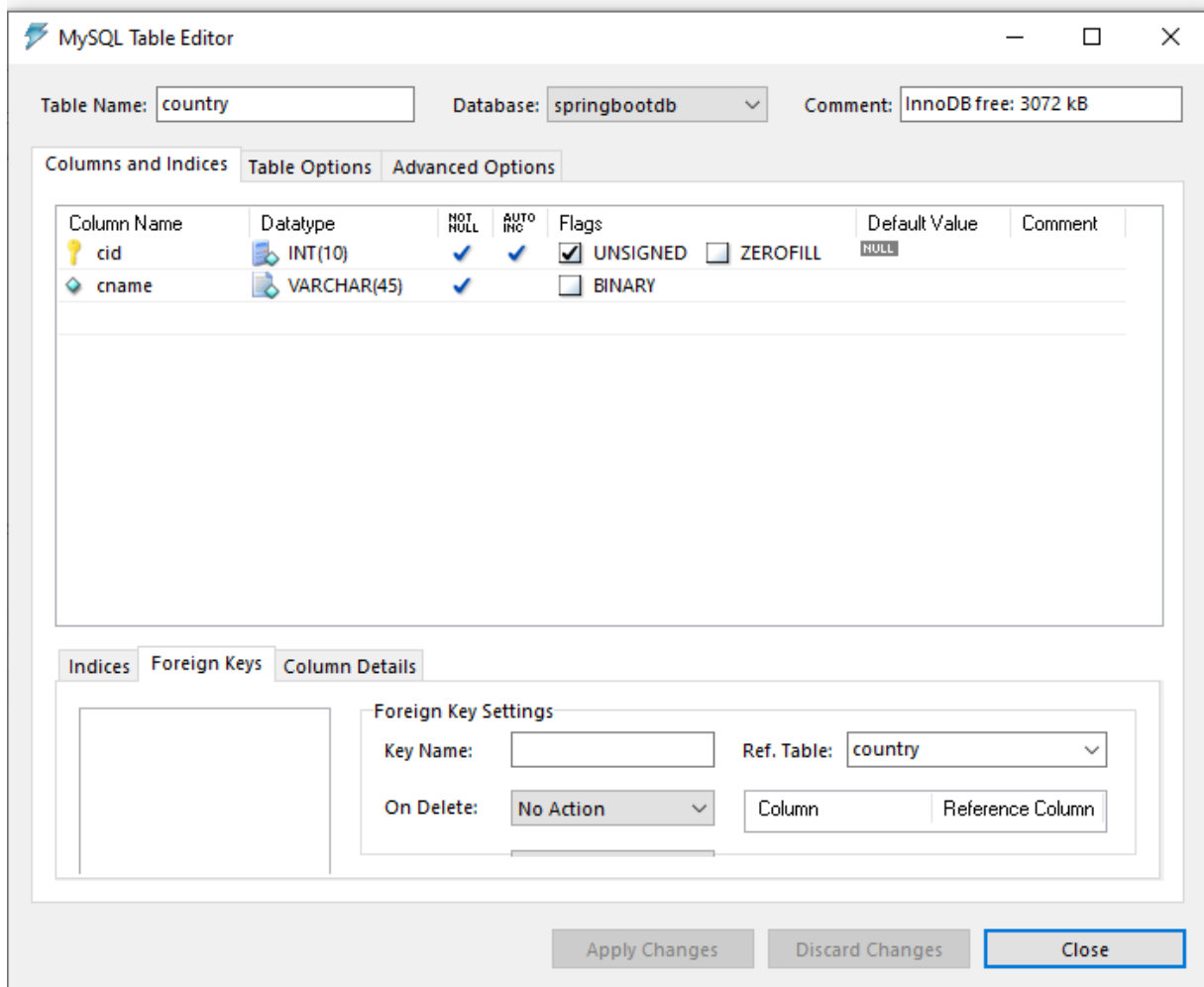
Foreign Key Settings

Key Name: Ref. Table:

On Delete: Column: Reference Column:

ESTD 2005

Table Structure:- Country







The image shows the MySQL Table Editor window for a table named 'country' in the 'springbootdb' database. The window has three tabs: 'Columns and Indices', 'Table Options', and 'Advanced Options'. The 'Columns and Indices' tab is active, showing a table with two columns: 'cid' (INT(10), UNSIGNED, NOT NULL, AUTO INCREMENT) and 'cname' (VARCHAR(45), BINARY). Below the table, there are tabs for 'Indices', 'Foreign Keys', and 'Column Details'. The 'Foreign Keys' tab is active, showing a 'Foreign Key Settings' section with a 'Key Name' field, a 'Ref. Table' dropdown set to 'country', and an 'On Delete' dropdown set to 'No Action'. There are also 'Column' and 'Reference Column' fields. At the bottom of the window, there are three buttons: 'Apply Changes', 'Discard Changes', and 'Close'.

MySQL Table Editor

Table Name: Database: Comment:

Columns and Indices Table Options Advanced Options

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
 cid	 INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	<input type="text" value="NULL"/>	
 cname	 VARCHAR(45)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> BINARY		

Indices Foreign Keys Column Details

Foreign Key Settings

Key Name: Ref. Table:

On Delete: Column Reference Column

Apply Changes Discard Changes Close

Employee.java

```
package com.jbk;
// Generated Jul 15, 2020 8:06:23 PM by Hibernate Tools 5.2.3.Final

import java.util.Date;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;

/**
 * Employee generated by hbm2java
 */
@Entity
@Table(name = "employee", catalog = "springbootdb")
public class Employee implements java.io.Serializable {

    private Integer id;
    private Country country;
    private String name;
    private String phoneno;
    private String departmentit;
    private String status;
    private Date createddtm;
    private String createdby;
    private Date updateddtm;
    private String updatedby;
```

```
public Employee() {  
}  
  
public Employee(Country country, Date createddtm, Date updateddtm) {  
    this.country = country;  
    this.createddtm = createddtm;  
    this.updateddtm = updateddtm;  
}  
  
public Employee(Country country, String name, String phoneno, String  
departmentit, String status, Date createddtm,  
                String createdby, Date updateddtm, String updatedby) {  
    this.country = country;  
    this.name = name;  
    this.phoneno = phoneno;  
    this.departmentit = departmentit;  
    this.status = status;  
    this.createddtm = createddtm;  
    this.createdby = createdby;  
    this.updateddtm = updateddtm;  
    this.updatedby = updatedby;  
}  
  
@Id  
@GeneratedValue(strategy = IDENTITY)  
  
@Column(name = "id", unique = true, nullable = false)  
public Integer getId() {  
    return this.id;  
}  
  
public void setId(Integer id) {  
    this.id = id;  
}  
  
@ManyToOne(fetch = FetchType.EAGER)  
@JoinColumn(name = "cid", nullable = false)  
public Country getCountry() {  
    return this.country;  
}
```

```
}

public void setCountry(Country country) {
    this.country = country;
}

@Column(name = "name", length = 45)
public String getName() {
    return this.name;
}

public void setName(String name) {
    this.name = name;
}

@Column(name = "phoneno", length = 45)
public String getPhoneno() {
    return this.phoneno;
}

public void setPhoneno(String phoneno) {
    this.phoneno = phoneno;
}

@Column(name = "departmentit", length = 45)
public String getDepartmentit() {
    return this.departmentit;
}

public void setDepartmentit(String departmentit) {
    this.departmentit = departmentit;
}

@Column(name = "status", length = 45)
public String getStatus() {
    return this.status;
}

public void setStatus(String status) {
    this.status = status;
}
```

```
}

@Temporal(TemporalType.TIMESTAMP)
@Column(name = "createddtm", nullable = false, length = 19)
public Date getCreateddtm() {
    return this.createddtm;
}

public void setCreateddtm(Date createddtm) {
    this.createddtm = createddtm;
}

@Column(name = "createdby", length = 45)
public String getCreatedby() {
    return this.createdby;
}

public void setCreatedby(String createdby) {
    this.createdby = createdby;
}

@Temporal(TemporalType.TIMESTAMP)
@Column(name = "updateddtm", nullable = false, length = 19)
public Date getUpdateddtm() {
    return this.updateddtm;
}

public void setUpdateddtm(Date updateddtm) {
    this.updateddtm = updateddtm;
}

@Column(name = "updatedby", length = 45)
public String getUpdatedby() {
    return this.updatedby;
}

public void setUpdatedby(String updatedby) {
    this.updatedby = updatedby;
}
```



```
}
```

Country.java

```
import java.util.HashSet;
import java.util.Set;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import javax.persistence.Table;

/**
 * Country generated by hbm2java
 */
@Entity
@Table(name = "country", catalog = "springbootdb")
public class Country implements java.io.Serializable {

    private Integer cid;
    private String cname;
    //private Set employees = new HashSet(0);

    public Country() {
    }

    public Country(String cname) {
        this.cname = cname;
    }

    /**
     public Country(String cname, Set employees) {
         this.cname = cname;
         this.employees = employees;
     }
    */
}
```

```
@Id
@GeneratedValue(strategy = IDENTITY)

@Column(name = "cid", unique = true, nullable = false)
public Integer getCid() {
    return this.cid;
}

public void setCid(Integer cid) {
    this.cid = cid;
}

@Column(name = "cname", nullable = false, length = 45)
public String getCname() {
    return this.cname;
}

public void setCname(String cname) {
    this.cname = cname;
}

/*
@OneToMany(fetch = FetchType.LAZY, mappedBy = "country")
public Set getEmployees() {
    return this.employees;
}

public void setEmployees(Set employees) {
    this.employees = employees;
}

*/
}
```


Program 1: Load Data

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

public class LoadData {

    public static void main(String[] args) {
        Configuration cfg=new Configuration();

        cfg.configure().addAnnotatedClass(Employee.class).addAnnotatedClass(Co
untry.class);

        SessionFactory sf=cfg.buildSessionFactory();
        Session session=sf.openSession();
        System.out.println("Employee Data");
        Employee emp=session.get(Employee.class, 1);
        System.out.println(emp.getCountry().getCid());
        System.out.println(emp.getCountry().getCname());
        System.out.println(emp.getName());
        System.out.println(emp.getPhoneno());
        System.out.println(emp.getDepartmentit());
        System.out.println(emp.getCreatedby());
        System.out.println(emp.getUpdatedby());
        /*
        Criteria criteria=session.createCriteria(Employee.class);
        List<Employee> listemployee=criteria.list();
        for(Employee emp:listemployee)
        {
            System.out.println(emp.getName());
            System.out.println(emp.getCountry().getCid());
            System.out.println(emp.getCountry().getCname());
        }
        */
    }
}
```

Program 2: Save Data

```
package com.jbk;
```

```
import java.time.LocalDateTime;
```

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

```
import org.hibernate.Session;
```

```
import org.hibernate.SessionFactory;
```

```
import org.hibernate.Transaction;
```

```
import org.hibernate.cfg.Configuration;
```

```
public class SaveData {
```

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

```
        Configuration cfg=new Configuration();
```

```
        cfg.configure().addAnnotatedClass(Employee.class).addAnnotatedClass(Co  
untry.class);
```

```
        SessionFactory sf=cfg.buildSessionFactory();
```

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

```
        Transaction tx=session.beginTransaction();
```

```
        Country ctr=session.get(Country.class, 1);
```

```
        ctr.getCid();
```

```
        Employee emp=new Employee();
```

```
        emp.setId(10);
```

```
        emp.setName("Abc");
```

```
        emp.setDepartmentit("Account");
```

```
        emp.setPhoneno("789684758");
```

```
        emp.setStatus("active");
```

```
        Date date=new Date();
```

```
        System.out.println(date);
```

```
        emp.setCreateddtm(date);
```

```
        emp.setCreatedby("Kiran");
```

```
        Date date1=new Date();
```

```
        emp.setUpdateddtm(date1);
```

```
emp.setUpdatedby("Kiran");  
emp.setCountry(ctr);
```

```
session.save(emp);  
tx.commit();
```

```
}  
}
```



Program 3: Update Data

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

public class UpdateData {

    public static void main(String[] args) {

        Configuration cfg=new Configuration();

        cfg.configure().addAnnotatedClass(Employee.class).addAnnotatedClass(Co
untry.class);

        SessionFactory sf=cfg.buildSessionFactory();
        Session session=sf.openSession();
        Transaction tx=session.beginTransaction();

        Employee employee=session.get(Employee.class,47);
        employee.setDepartmentit("Marketing");

        session.update(employee);

        tx.commit();

        System.out.println("Employee Updated Successfully");
    }
}
```

Program 4: Save or Update

```
import java.util.Date;

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

public class SaveOrUpdateData {

    public static void main(String[] args) {
        Configuration cfg=new Configuration();

        cfg.configure().addAnnotatedClass(Employee.class).addAnnotatedClass(Co
untry.class);

        SessionFactory sf=cfg.buildSessionFactory();
        Session session=sf.openSession();
        Transaction tx=session.beginTransaction();
        Country ctr=session.get(Country.class, 1);
        ctr.getCid();

        Employee emp=new Employee();
        //emp.setId(47);
        emp.setName("Abc");
        emp.setDepartmentit("Marketing");
        emp.setPhoneno("789684758");
        emp.setStatus("active");
        Date date=new Date();
        System.out.println(date);
        emp.setCreateddtm(date);
        emp.setCreatedby("Kiran");
        Date date1=new Date();
        emp.setUpdateddtm(date1);
        emp.setUpdatedby("Kiran");
        emp.setCountry(ctr);
```

```
        session.saveOrUpdate(emp);  
        tx.commit();  
        System.out.println("Record updated");  
    }  
}
```



Program 5: Delete Data

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

public class DeleteData {

    public static void main(String[] args) {
        Configuration cfg=new Configuration();

        cfg.configure().addAnnotatedClass(Employee.class).addAnnotatedClass(Co
untry.class);

        SessionFactory sf=cfg.buildSessionFactory();
        Session session=sf.openSession();
        Transaction tx=session.beginTransaction();

        Employee emp=session.get(Employee.class, 46);
        session.delete(emp);
        tx.commit();
        System.out.println("Employee Deleted Successfully");
    }
}
```