

## DB 实验 4：数据库应用系统开发

### 目录

一、 实验目的及要求.....	1
(1) 实验目的.....	1
(2) 实验要求.....	1
二、 开发工具 .....	2
三、 数据表.....	2
四、 前台应用程序 .....	3
五、 查询与展示.....	4
六、 插删改操作.....	4
七、 代码.....	6
(1) Entity .....	6
(2) Mapper.....	7
(3) MapperXml .....	8
(4) Service .....	8
(5) ServiceImpl.....	9
(6) Controller.....	10
(7) Add html.....	12
(8) Edit html.....	14
(9) Index html .....	16
(10) Application.....	19

## 一、 实验目的及要求

### (1) 实验目的

设计一个简单的应用系统，熟练掌握 MySQL 基本命令、SQL 语言，学习简单数据库系统的设计方法，包括数据库概要设计、逻辑设计。

### (2) 实验要求

- 定义至少两张数据表，表中不少于 10 条记录

- 能够利用前台应用程序正常地连接数据库
- 能够根据输入条件查询表中内容，并用表格有效展示
- 能够实现对基础数据进行插删改操作

## 二、开发工具

- 数据库平台：mysql + workbench
- 前台开发工具：SpringBoot + MyBatis plus + Thymeleaf
- 高级语言：java

## 三、数据表

本实验参考[实验三：数据库系统开发](#)设计了 10 个数据表，总体 ER 图如下：

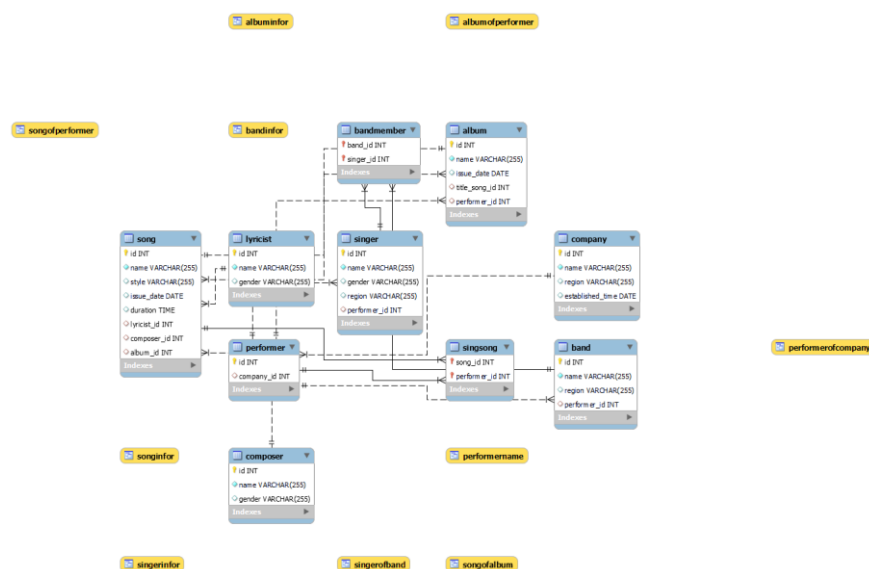


图 1 数据表总体 ER 图

将 ER 图转化为关系模式如下，带下划线的属性为主键，加粗的为非空属性，斜体的为外键。

### 实体集

Song(id, **name**, style, issue\_date, duration, *lyricist\_id*, *composer\_id*, *album\_id*)  
 Album(id, **name**, issue\_date, *title\_song\_id*, *performer\_id*)  
 Performer(id, *company\_id*)  
 Band(id, **name**, region, *performer\_id*)  
 Singer(id, **name**, gender, region, *performer\_id*)  
 Company(id, **name**, region, established\_time)  
 Lyricist(id, **name**, gender)  
 Composer(id, **name**, gender)

### 联系集

SingSong(song\_id, performer\_id)

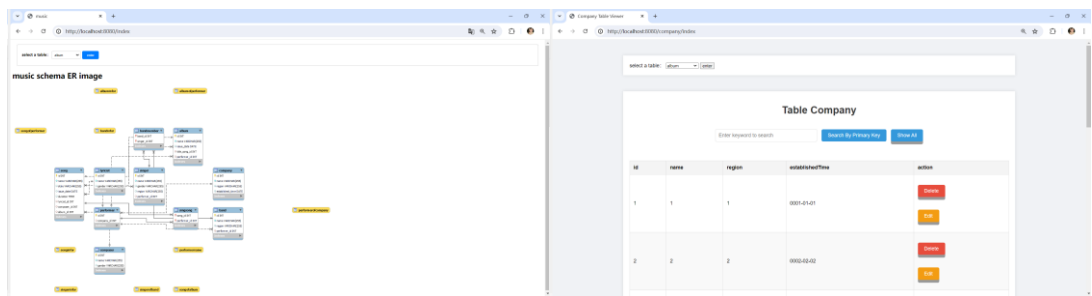
BandMember(band\_id, singer\_id)

每个数据表中存储 10 条从 1-10 的记录，如图。

album	band	bandmember	company	composer	lyricist	performer	singer	singsong	song
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9
10	10	10	10	10	10	10	10	10	10

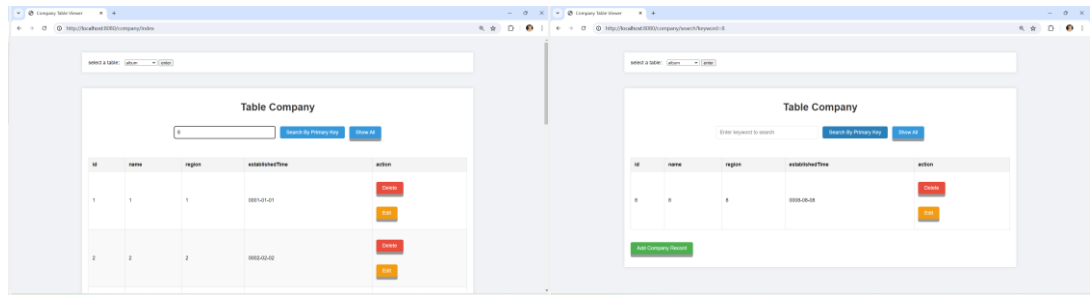
## 四、 前台应用程序

在命令行中使用 `gradlew build`、`gradlew bootRun`，浏览器进入 <http://localhost:8080/index>，即可访问本数据库应用系统。进入之后可以选择数据表，点击 enter 即可看到数据表，可对数据进行增删改查，我们这里以数据表 `company` 为例。



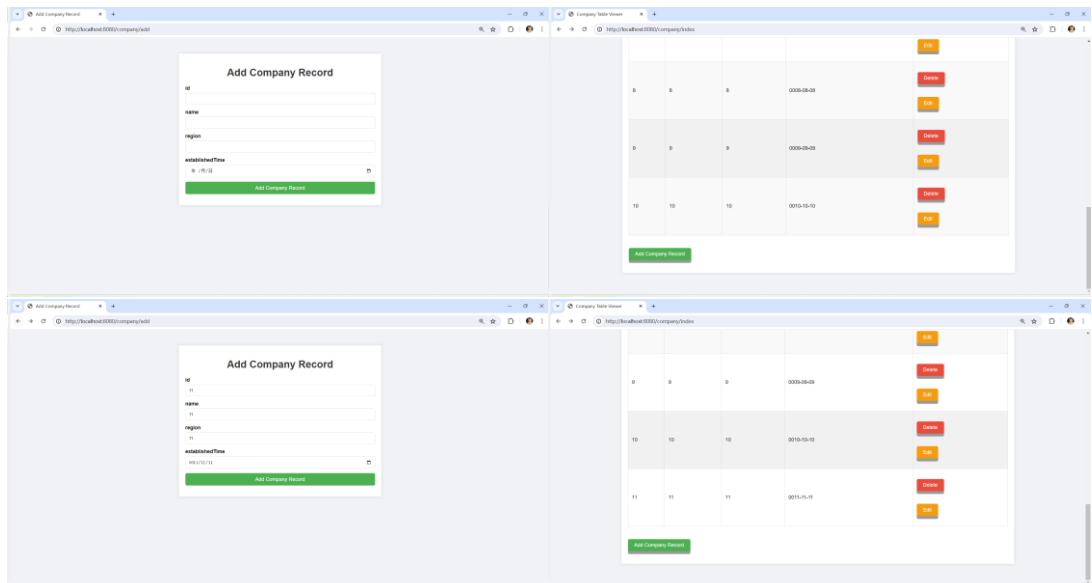
## 五、 查询与展示

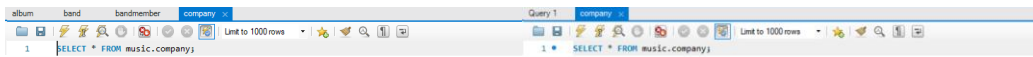
由于每个数据表中都有很多属性，根据部分属性查询数据的难度较高，所以这里我们主要采用数据表的主键进行数据查询，对于联系集 `BandMember` 和 `SingSong` 我们则根据它们的两个外键进行数据查询，下面给出 `company` 的查询示例。



## 六、 插删改操作

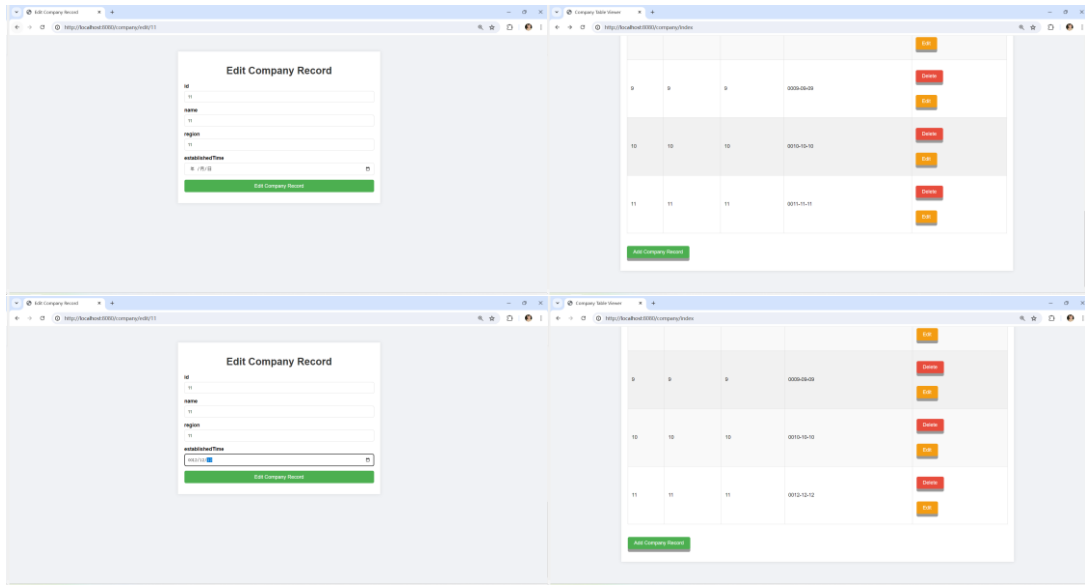
以 `company` 为例，**插入数据**的界面如图，插入前有 10 条数据，我们这里插入第 11 条数据，插入结果如图。





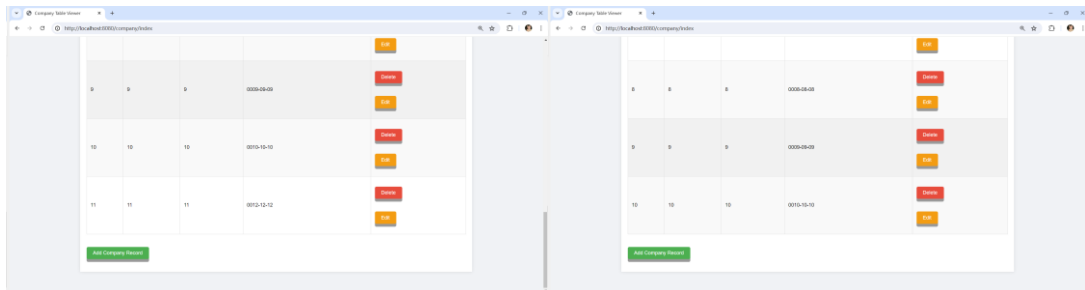
id	name	region	established_time
1	1	1	0001-01-01
2	2	2	0002-02-02
3	3	3	0003-03-03
4	4	4	0004-04-04
5	5	5	0005-05-05
6	6	6	0006-06-06
7	7	7	0007-07-07
8	8	8	0008-08-08
9	9	9	0009-09-09
10	10	10	0010-10-10

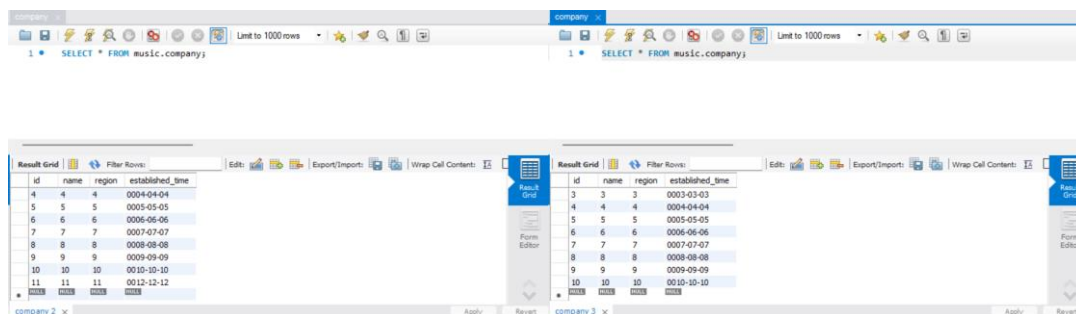
接下来，我们进行**修改操作**，将第 11 条数据的时间修改为 12/12/12，修改结果如图。



id	name	region	established_time
4	4	4	0004-04-04
5	5	5	0005-05-05
6	6	6	0006-06-06
7	7	7	0007-07-07
8	8	8	0008-08-08
9	9	9	0009-09-09
10	10	10	0010-10-10
11	11	11	0012-12-12

最后，进行**删除操作**，点击 delete 即可。





## 七、代码

由于本系统有 10 个数据表，代码太多，我们这里以其中一个表为例，附该表相关的所有代码，全部代码可查看 [music-database-application-system](#)。

### (1) Entity

```
package com.example.music.entity;

import com.baomidou.mybatisplus.annotation.IdType;
import com.baomidou.mybatisplus.annotation.TableId;
import java.io.Serializable;
import java.time.LocalDate;
import io.swagger.annotations.ApiModel;
import io.swagger.annotations.ApiModelProperty;

/**
 * <p>
 *
 * </p>
 *
 * @author cxlhyx
 * @since 2024-5-31
 */
@ApiModel(value = "Company 对象", description = "")
public class Company implements Serializable {

    private static final long serialVersionUID = 1L;

    @TableId(value = "id", type = IdType.AUTO)
    private Integer id;

    private String name;

    private String region;
```

```

private LocalDate establishedTime;

public Integer getId() {
    return id;
}

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

public String getName() {
    return name;
}

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

public String getRegion() {
    return region;
}

public void setRegion(String region) {
    this.region = region;
}

public LocalDate getEstablishedTime() {
    return establishedTime;
}

public void setEstablishedTime(LocalDate establishedTime) {
    this.establishedTime = establishedTime;
}

@Override
public String toString() {
    return "Company{" +
        "id=" + id +
        ", name=" + name +
        ", region=" + region +
        ", establishedTime=" + establishedTime +
        "}";
}
}

```

## (2) Mapper

```
package com.example.music.mapper;
```

```

import com.example.music.entity.Company;
import com.baomidou.mybatisplus.core.mapper.BaseMapper;

import java.util.List;
import java.util.Map;

/**
 * <p>
 * Mapper 接口
 * </p>
 *
 * @author cxlhyx
 * @since 2024-5-31
 */
public interface CompanyMapper extends BaseMapper<Company> {
    /**
     * 获取 Company 全部数据
     */
    public List<Company> getAll(Map<String, Object> paramMap);

    /**
     * 根据 id 删除 Company 数据
     * @param id
     */
    public void delete(Integer id);
}

```

### (3) MapperXml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.example.music.mapper.CompanyMapper">
    <select id="getAll" resultType="com.example.music.entity.Company">
        SELECT id, name, region, established_time As 'establishedTime'
    FROM company;
    </select>
    <delete id="delete">
        DELETE FROM company WHERE id=#{id};
    </delete>
</mapper>

```

### (4) Service

```

package com.example.music.service;

```



```

import com.example.music.entity.Company;
import com.baomidou.mybatisplus.extension.service.IService;

import java.util.List;
import java.util.Map;

/**
 * <p>
 * 服务类
 * </p>
 *
 * @author cxlhyx
 * @since 2024-5-31
 */
public interface ICompanyService extends IService<Company> {
    /**
     * 获取 company 全部数据
     */
    public List<Company> getAll(Map<String, Object> paramMap);

    /**
     * 根据 id 删除 company 数据
     *
     * @param id
     */
    public void delete(Integer id);
}

```

## (5) ServiceImpl

```

package com.example.music.service.impl;

import com.example.music.entity.Company;
import com.example.music.mapper.CompanyMapper;
import com.example.music.service.ICompanyService;
import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.Map;

/**
 * <p>

```

```

* 服务实现类
* </p>
*
* @author cxlhyx
* @since 2024-5-31
*/
@Service
public class CompanyServiceImpl extends ServiceImpl<CompanyMapper,
Company> implements ICompanyService {
    @Autowired
    private CompanyMapper companyMapper;

    @Override
    public List<Company> getAll(Map<String, Object> paramMap) {
        return companyMapper.getAll(paramMap);
    }

    @Override
    public void delete(Integer id) {
        companyMapper.delete(id);
    }
}

```

## (6) Controller

```

package com.example.music.controller;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.example.music.entity.Company;
import com.example.music.service.ICompanyService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

import org.springframework.stereotype.Controller;

import java.time.LocalDate;

/**
* <p>
* 前端控制器
* </p>
*
* @author cxlhyx

```

```

    * @since 2024-5-31
    */
@Controller
@RequestMapping("/company")
public class CompanyController {
    private String prefix = "company/";
    @Autowired
    private ICompanyService iCompanyService;

    private static final Logger logger =
LoggerFactory.getLogger(CompanyController.class);

    @RequestMapping("/index")
    public String index(Model model) {
        model.addAttribute("companys",
iCompanyService.getAll(model.asMap()));
        logger.info("company/index");
        return prefix + "indexCompany.html";
    }

    @GetMapping("/add")
    public String showAdd() {
        logger.info("company/add");
        return prefix + "addCompany.html";
    }

    @PostMapping("/add")
    public String add(@RequestParam("id") Integer id,
        @RequestParam("name") String name,
        @RequestParam("region") String region,
        @RequestParam("establishedTime") LocalDate
establishedTime) {
        Company company = new Company();
        company.setId(id);
        company.setName(name);
        company.setRegion(region);
        company.setEstablishedTime(establishedTime);
        iCompanyService.save(company);
        logger.info("company/add " + company);
        return "redirect:/" + prefix + "index";
    }

    @PostMapping("/delete/{id}")
    public String delete(@PathVariable("id") Integer id) {

```

```

        iCompanyService.delete(id);
        logger.info("company/delete " + id);
        return "redirect:/" + prefix + "index";
    }

    @GetMapping("/edit/{id}")
    public String showEdit(@PathVariable("id") Integer id, Model model)
    {
        model.addAttribute("company", iCompanyService.getById(id));
        logger.info("company/edit " + id);
        return prefix + "editCompany.html";
    }

    @PostMapping("/edit")
    public String edit(@RequestParam("id") Integer id,
                      @RequestParam("name") String name,
                      @RequestParam("region") String region,
                      @RequestParam("establishedTime") LocalDate
establishedTime) {
        Company company = new Company();
        company.setId(id);
        company.setName(name);
        company.setRegion(region);
        company.setEstablishedTime(establishedTime);
        iCompanyService.updateById(company);
        logger.info("company/edit " + id + " " + company);
        return "redirect:/" + prefix + "index";
    }

    @GetMapping("/search")
    public String searchRecords(@RequestParam("keyword") String keyword,
Model model) {
        model.addAttribute("companys",
iCompanyService.getById(keyword));
        logger.info("company/search " + keyword);
        return prefix + "indexCompany.html";
    }
}

```

## (7) Add html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">

```

```
<title>Add Company Record</title>
<style>
  body {
    font-family: 'Arial', sans-serif;
    background-color: #f0f2f5;
    margin: 0;
    padding: 0;
  }
  .container {
    max-width: 600px;
    margin: 50px auto;
    padding: 20px;
    background-color: #fff;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }
  h1 {
    text-align: center;
    color: #333;
  }
  form {
    display: flex;
    flex-direction: column;
  }
  label {
    margin-bottom: 5px;
    font-weight: bold;
  }
  input, button {
    padding: 10px;
    margin-bottom: 15px;
    border: 1px solid #ddd;
    border-radius: 5px;
  }
  button {
    background-color: #4CAF50;
    color: #fff;
    cursor: pointer;
    border: none;
    font-size: 16px;
  }
  button:hover {
    background-color: #45a049;
  }
</style>
```

```

</head>
<body>
<div class="container">
  <h1>Add Company Record</h1>
  <form action="/company/add" method="post">
    <label for="id">id</label>
    <input type="number" id="id" name="id" required>

    <label for="name">name</label>
    <input type="text" id="name" name="name" required>

    <label for="region">region</label>
    <input type="text" id="region" name="region">

    <label for="establishedTime">establishedTime</label>
    <input type="date" id="establishedTime" name="establishedTime">

    <button type="submit">Add Company Record</button>
  </form>
</div>
</body>
</html>

```

## (8) Edit html

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Edit Company Record</title>
  <style>
    body {
      font-family: 'Arial', sans-serif;
      background-color: #f0f2f5;
      margin: 0;
      padding: 0;
    }
    .container {
      max-width: 600px;
      margin: 50px auto;
      padding: 20px;
      background-color: #fff;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }
    h1 {

```

```

        text-align: center;
        color: #333;
    }
    form {
        display: flex;
        flex-direction: column;
    }
    label {
        margin-bottom: 5px;
        font-weight: bold;
    }
    input, button {
        padding: 10px;
        margin-bottom: 15px;
        border: 1px solid #ddd;
        border-radius: 5px;
    }
    button {
        background-color: #4CAF50;
        color: #fff;
        cursor: pointer;
        border: none;
        font-size: 16px;
    }
    button:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>
<div class="container">
    <h1>Edit Company Record</h1>
    <form action="/company/edit" th:object="${company}" method="post">
        <label for="id">id</label>
        <input type="number" id="id" th:field="*{id}" required>

        <label for="name">name</label>
        <input type="text" id="name" th:field="*{name}" required>

        <label for="region">region</label>
        <input type="text" id="region" th:field="*{region}">

        <label for="establishedTime">establishedTime</label>

```

```

        <input type="date" id="establishedTime"
th:field="*{establishedTime}">

        <button type="submit">Edit Company Record</button>
    </form>
</div>
</body>
</html>

```

## (9) Index.html

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Company Table Viewer</title>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            background-color: #f0f2f5;
            margin: 0;
            padding: 0;
        }
        .container {
            max-width: 1200px;
            margin: 50px auto;
            padding: 20px;
            background-color: #fff;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }
        h1 {
            text-align: center;
            color: #333;
        }
        .table-container {
            overflow-x: auto;
        }
        table {
            width: 100%;
            border-collapse: collapse;
            margin: 20px 0;
        }
        table, th, td {
            border: 1px solid #ddd;
        }
    </style>

```



```
th, td {
    padding: 12px;
    text-align: left;
}
th {
    background-color: #f4f4f4;
}
tr:nth-child(even) {
    background-color: #f9f9f9;
}
tr:hover {
    background-color: #f1f1f1;
}
.button {
    display: inline-block;
    padding: 10px 20px;
    font-size: 16px;
    cursor: pointer;
    text-align: center;
    text-decoration: none;
    outline: none;
    color: #fff;
    background-color: #4CAF50;
    border: none;
    border-radius: 5px;
    box-shadow: 0 9px #999;
    margin: 20px 0;
}
.button:hover {
    background-color: #45a049;
}
.button:active {
    background-color: #45a049;
    box-shadow: 0 5px #666;
    transform: translateY(4px);
}
.delete-button {
    background-color: #e74c3c;
}
.delete-button:hover {
    background-color: #c0392b;
}
.edit-button {
    background-color: #f39c12;
```

```

    }
    .edit-button:hover {
        background-color: #e67e22;
    }
    .search-container {
        text-align: center;
        margin-bottom: 20px;
    }
    .search-input {
        padding: 10px;
        font-size: 16px;
        width: 300px;
        border: 1px solid #ddd;
        border-radius: 5px;
    }
    .search-button, .show-all-button {
        padding: 10px 20px;
        font-size: 16px;
        cursor: pointer;
        text-align: center;
        text-decoration: none;
        outline: none;
        color: #fff;
        background-color: #3498db;
        border: none;
        border-radius: 5px;
        margin-left: 10px;
    }
    .search-button:hover, .show-all-button:hover {
        background-color: #2980b9;
    }
</style>
</head>
<body>
<div th:insert=~{tableSelect :: content}></div>
<div class="container">
    <h1>Table Company</h1>
    <div class="search-container">
        <form th:action="@{/company/search}" method="get">
            <input type="text" name="keyword" class="search-input"
placeholder="Enter keyword to search" required>
            <button type="submit" class="search-button">Search By
Primary Key</button>

```

```

        <a href="/company/index" class="button show-all-button">Show
All</a>
    </form>
</div>
<div class="table-container">
    <table>
        <tbody>
            <thead>
                <tr>
                    <th>id</th>
                    <th>name</th>
                    <th>region</th>
                    <th>establishedTime</th>
                    <th>action</th>
                </tr>
            </thead>
            <tr th:each="company : ${companys}">
                <td th:text="${company.id}"></td>
                <td th:text="${company.name}"></td>
                <td th:text="${company.region}"></td>
                <td th:text="${company.establishedTime}"></td>
                <td>
                    <form
th:action="@{/company/delete/{id}(id=${company.id})}" method="post">
                        <button type="submit" class="button delete-
button">Delete</button>
                    </form>
                    <a th:href="@{/company/edit/{id}(id=${company.id})}"
class="button edit-button">Edit</a>
                </td>
            </tr>
        </tbody>
    </table>
</div>
<a href="/company/add" class="button">Add Company Record</a>
</div>
</body>
</html>

```

## (10) Application

```

package com.example.music;

import org.mybatis.spring.annotation.MapperScan;
import org.springframework.boot.SpringApplication;

```

```
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.core.env.Environment;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

@SpringBootApplication
@RestController
@MapperScan("com.example.music.mapper")    //指定 Mapper 接口所在的包,将它们
自动注册为 Spring Bean,无需每个 Mapper 接口都单独使用@Mapper 注解
public class MusicApplication {

    public static void main(String[] args) {
        ConfigurableApplicationContext run =
SpringApplication.run(MusicApplication.class, args);
        Environment env = run.getEnvironment();
        System.out.println("The system started successfully...");
        System.out.println("System access address:http://localhost:" +
env.getProperty("server.port") + "/index");
    }

    @GetMapping("/hello")
    public String hello(@RequestParam(value = "name", defaultValue =
"World") String name) {
        return String.format("Hello %s!", name);
    }
}
```