

数据库大作业

任务一实验报告

邓奇恩 78066014, 谢宇隆 78066015, 苏嘉娴 78066009

1. 开发环境与技术说明

程序分为前端和后端。前端使用了 React 框架，开发语言是 JavaScript。后端使用的是 Express 框架，开发语言是 Node.js。使用的数据库管理系统是 MySQL。

开发过程中使用了前后端分离的方法，方便开发调试。前端地址在

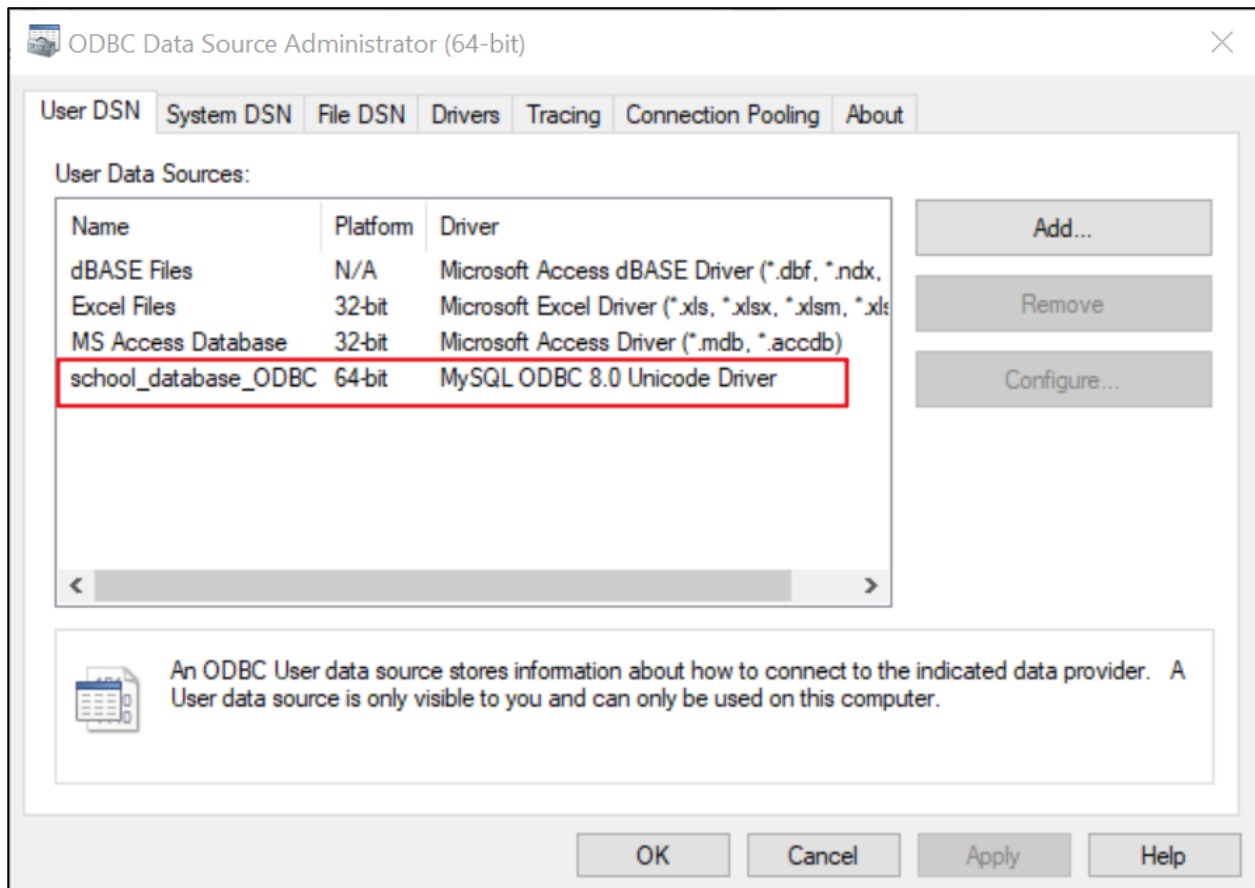
<http://localhost:3000/>，后端地址在 <http://localhost:3001/>。

前端通过特定的 API 地址向后端发送请求，而后端通过 npm-odbc 模块连接 ODBC 管理器，向本地的 MySQL 数据库发送请求。

使用方式：打开两个命令行，一个 path 是前端文件夹的主目录，一个 path 是后端文件夹的主目录。然后在两个命令行分别输入“npm start”，程序就可以在浏览器上运行了。

程序连接 ODBC 的方式如下：

在 ODBC 管理器里，User DSN 是“school_database_ODBC”，User 是“root”，Password 是“password”。



MySQL Connector/ODBC Data Source Configuration

MySQL Connector/ODBC

Connection Parameters

Data Source Name: school_database_ODBC

Description:

☒ TCP/IP Server: LOCALHOST Port: 3306

☐ Named Pipe:

User: root

Password: ●●●●●●●●

Database: school_database ▼ Test

Details >> OK Cancel Help

数据库后端.env 文件里定义了 ODBC_CONNECTION_STRING。

ODBC_CONNECTION_STRING="DSN=school_database_ODBC;Uid=root;Pwd=password;"

在后端会使用这个连接字符串通过 ODBC API 连接上 ODBC 管理器。

2. 数据库表的定义

在本地 MySQL 的数据库名是 school_database, 数据库内定义了 3 个表 student、teacher 和 course。三个表的建立方式定义如下：

```
CREATE TABLE student(  
  student_id VARCHAR(100),  
  name VARCHAR(100),
```

```

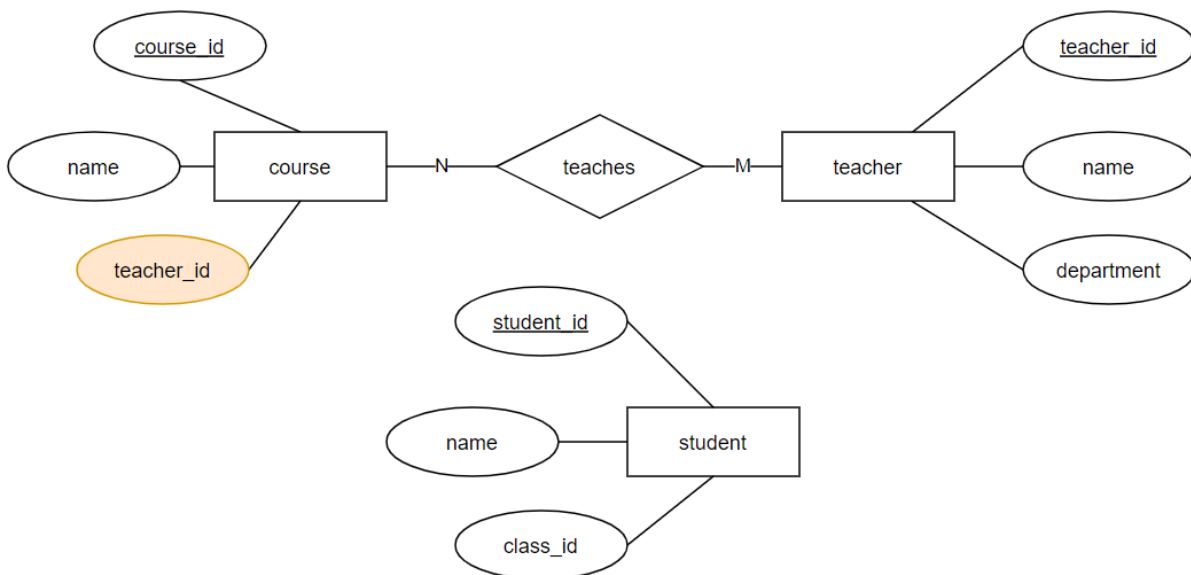
class_id VARCHAR(100),
PRIMARY KEY(student_id)
);

CREATE TABLE teacher(
    teacher_id VARCHAR(100),
    name VARCHAR(100),
    department VARCHAR(100),
    PRIMARY KEY(teacher_id)
);

CREATE TABLE course(
    course_id VARCHAR(100),
    name VARCHAR(100),
    teacher_id VARCHAR(100),
    PRIMARY KEY(course_id, teacher_id),
    FOREIGN KEY(teacher_id) REFERENCES teacher(teacher_id) ON DELETE C
ASCAD
);

```

数据库的 E-R 图如下：



3. 数据库的连接方法与语句，实现增、删、改、查等功能的语句

以连接 course 表为例子。

程序与 odbc 的连接由后端负责。连接 odbc 的字符串 ODBC_CONNECTION_STRING 定义在.env 文件里。

ODBC_CONNECTION_STRING="DSN=school_database_ODBC;Uid=root;Pwd=password;"

若增加数据或修改数据时某属性为空，则向数据库请求的语句相应的属性为 NULL（而非字符串），对于这种情况暂时没找到比较优雅的处理方式，所以对于查询语句 queryStr 的判断较为冗长。

```
const teachersRouter = require('express').Router()

const odbc = require('odbc'); //使用 node-odbc 模块

teachersRouter.get('/', async (request, response) => { //查
  const connectionConfig = {
    connectionString: process.env.ODBC_CONNECTION_STRING,
  };
  const connection = await odbc.connect(connectionConfig);
  const result = await connection.query('SELECT * FROM teacher');
  response.send(JSON.stringify(result));
})

teachersRouter.post('/', async (request, response) => { //增
  const connectionConfig = {
    connectionString: process.env.ODBC_CONNECTION_STRING,
  };
  const teacher = request.body;
  try {
    const connection = await odbc.connect(connectionConfig);
    var queryStr = 'INSERT INTO teacher VALUES('
    if (teacher.teacher_id === '') {
      queryStr += `NULL, `;
    }
    else {
      queryStr += `${teacher.teacher_id}, `;
    }
    if (teacher.name === '') {
      queryStr += `NULL, `;
    }
    else {
      queryStr += `${teacher.name}, `;
    }
    if (teacher.department === '') {
      queryStr += `NULL)`;
    }
    else {
      queryStr += `${teacher.department})`;
    }
  }
}
```

```

    }
    await connection.query(queryStr);
    // await connection.query(`INSERT INTO teacher VALUES('${teacher.teacher_id}', '${teacher.name}', '${teacher.department}')`);
    const result = await connection.query(`SELECT ROW_COUNT()`);
    if (result.count === 1) {
        response.status(204).end();
    }
    else {
        response.status(404).json({error: `${JSON.stringify(result)}`});
    }
}
catch(err) {
    response.status(404).json({error: `${JSON.stringify(err.odbErrors)}`});
}
})

teachersRouter.delete('/:id', async (request, response) => { //删
    const connectionConfig = {
        connectionString: process.env.ODBC_CONNECTION_STRING,
    };
    const connection = await odbc.connect(connectionConfig);
    try {
        const id = request.params.id;
        await connection.query(`DELETE FROM teacher WHERE teacher_id='${id}'`);
        const result = await connection.query(`SELECT ROW_COUNT()`);
        if (result.count === 1) {
            response.status(204).end();
        }
        else {
            response.status(404).json({error: `${JSON.stringify(result)}`});
        }
    }
    catch(err) {
        response.status(404).json({error: `${JSON.stringify(err.odbErrors)}`});
    }
})

teachersRouter.put('/:id', async (request, response) => { //改
    const connectionConfig = {
        connectionString: process.env.ODBC_CONNECTION_STRING,
    };
    const connection = await odbc.connect(connectionConfig);
    try {
        const teacher = request.body;
        const id = request.params.id;
        var queryStr = 'UPDATE teacher SET ';
        if (teacher.teacher_id === '') {
            queryStr += 'teacher_id=NULL, ';
        }
    }

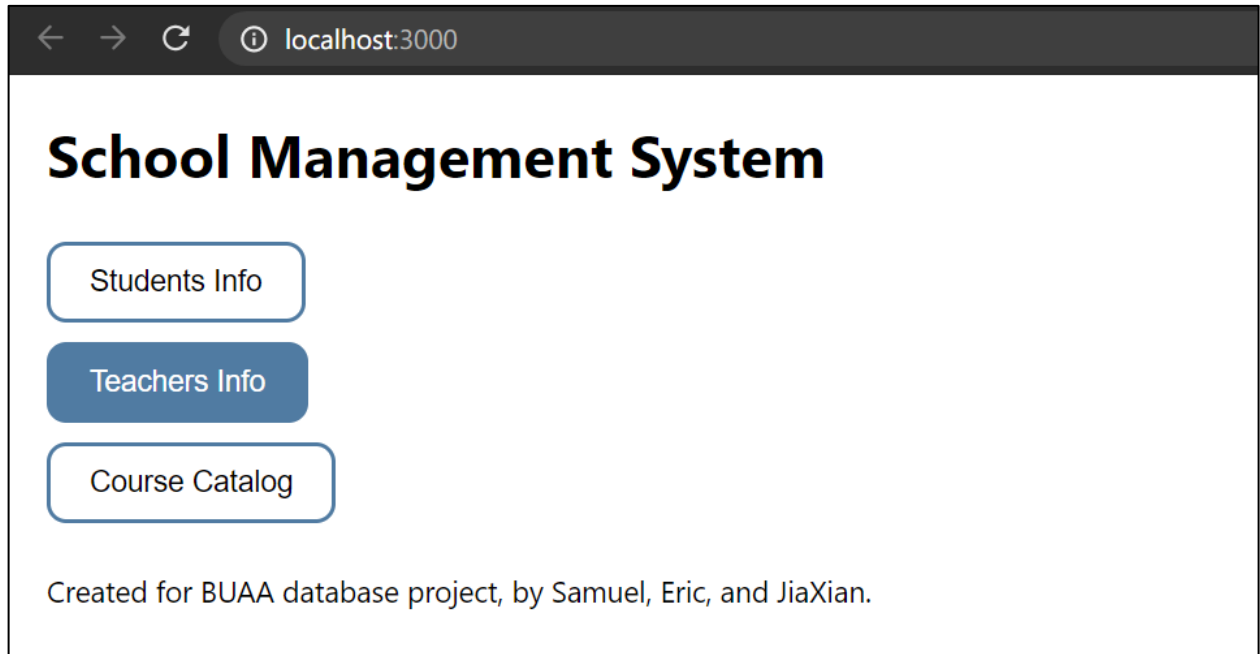
```

```

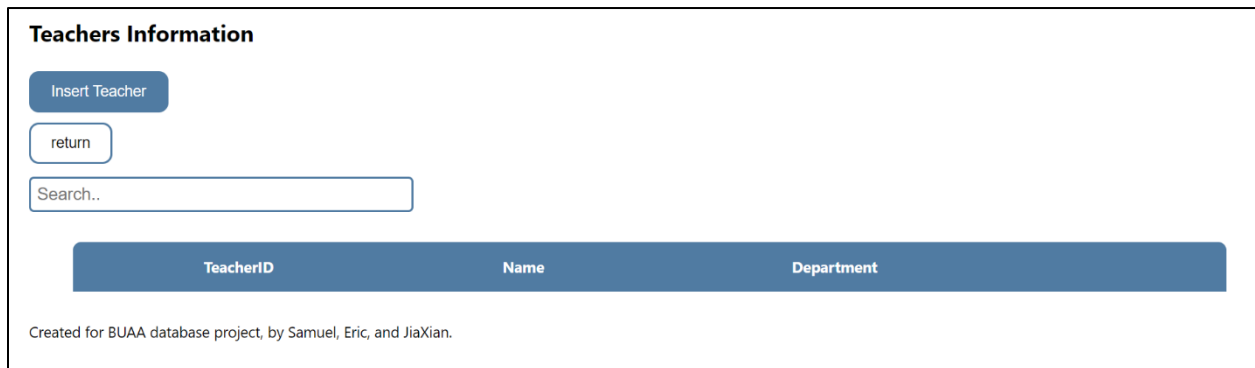
    else {
        queryStr += `teacher_id='${teacher.teacher_id}',` ;
    }
    if (teacher.name === '') {
        queryStr += 'name=NULL, ';
    }
    else {
        queryStr += `name='${teacher.name}',` ;
    }
    if (teacher.department === '') {
        queryStr += 'department=NULL, ';
    }
    else {
        queryStr += `department='${teacher.department}' ` ;
    }
    queryStr += `WHERE teacher_id='${id}';`;
    // await connection.query
    // (`UPDATE teacher
    //     SET teacher_id='${teacher.teacher_id}', name='${teacher.name}', departmen
t='${teacher.department}'
    //     WHERE teacher_id='${id}';`
    // );
    await connection.query(queryStr);
    const result = await connection.query(`SELECT ROW_COUNT()`);
    if (result.count === 1) {
        response.status(204).end();
    }
    else {
        response.status(404).json({error: `${JSON.stringify(result)}`});
    }
}
catch(err) {
    response.status(404).json({error: `${JSON.stringify(err.odbErrors)}`});
}
})

```

4. 展现各种功能实现结果的截图
 - a. 首页有三个按钮，对应着三个表。



- b. 点击 Teachers Info 按钮进入教师页面。暂时教师表为空。



c. 点击 Insert Teacher 按钮展开“输入教师”表格。在表内输入教师名字和院系。

Teachers Information

Insert Teacher

TeacherID

Name

Desmond

Department

CS

Cancel

Confirm

return

Search..

TeacherID	Name	Department
-----------	------	------------

d. 点击 confirm 确定提交后，跳出错误提示，表示 teacher_id 不能为 NULL 值。

Teachers Information

Insert Teacher

TeacherID

Name

Department

Cancel

Confirm

localhost:3000 says
insert failed
Error: [MySQL][ODBC 8.0(w) Driver][mysqld-8.0.21]Column 'teacher_id'
cannot be null

OK

- e. 再次填写表格并提交。成功插入后可看到 teacher 表中的表项出现在下方表格。

Teachers Information

Insert Teacher

TeacherID

1234

Name

Desmond

Department

CS

Cancel

Confirm

Search..

TeacherID	Name	Department		
1234	Desmond	CS	delete	update

Created for BUAA database project, by Samuel, Eric, and JiaXian.

- f. 再向表中增加几个教师项。

Search..

TeacherID	Name	Department		
1234	Desmond	CS	delete	update
J106	Wong	Electronic	delete	update
H123	Lee	Literature	delete	update

Created for BUAA database project, by Samuel, Eric, and JiaXian.

- g. 点击 **update** 按钮。把 J106 改为 J1064, Electronic 改为 EE。然后点击 **confirm**, 这时会出现确定框, 点击 **OK**。就修改成功了。

TeacherID	Name	Department		
1234	Desmond	CS	<button>delete</button>	<button>update</button>
<input type="text" value="J106"/>	Wong	Electronic	<button>cancel</button>	<button>confirm</button>
H123	Lee	Literature	<button>delete</button>	<button>update</button>

Department

Cancel

localhost:3000 says
confirm update?

OK

Cancel

turn

arch..

TeacherID	Name	Department		
1234	Desmond	CS	<button>delete</button>	<button>update</button>
<input type="text" value="J1064"/>	<input type="text" value="Wong"/>	<input type="text" value="EE"/>	<button>cancel</button>	<button>confirm</button>
H123	Lee	Literature	<button>delete</button>	<button>update</button>

TeacherID	Name	Department		
1234	Desmond	CS	<button>delete</button>	<button>update</button>
J1064	Wong	EE	<button>delete</button>	<button>update</button>
H123	Lee	Literature	<button>delete</button>	<button>update</button>

h. 若把第二个表项的 J1064 修改为空，则出现错误提示，修改失败。

Department

Cancel

localhost:3000 says
update failed
Error: [MySQL][ODBC 8.0(w) Driver][mysql-8.0.21]Column 'teacher_id'
cannot be null

OK

return

Search..

TeacherID	Name	Department		
1234	Desmond	CS	delete	update
	Wong	EE	cancel	confirm
H123	Lee	Literature	delete	update

i. 搜索框可以实时搜索匹配所有的表项内容。搜索字符串对大小写不敏感。

des

TeacherID	Name	Department		
1234	Desmond	CS	delete	update

ee

TeacherID	Name	Department		
H123	Lee	Literature	delete	update
J1064	Wong	EE	delete	update

j. 删除表项时点击 delete 按钮，然后在确定框中选 OK，就成功删除了。

Teachers Information

Insert Teacher

return

Search..

localhost:3000 says
confirm delete?

OK

Cancel

TeacherID	Name	Department		
1234	Desmond	CS	<div>delete</div>	<div>update</div>
H123	Lee	Literature	<div>delete</div>	<div>update</div>
J1064	Wong	EE	<div>delete</div>	<div>update</div>

TeacherID	Name	Department		
H123	Lee	Literature	<div>delete</div>	<div>update</div>
J1064	Wong	EE	<div>delete</div>	<div>update</div>

k. 点击 return 按钮返回首页。

Teachers Information

Insert Teacher

return

Search..

TeacherID	Name	Department		
H123	Lee	Literature	<div>delete</div>	<div>update</div>
J1064	Wong	EE	<div>delete</div>	<div>update</div>

Created for BUAA database project, by Samuel, Eric, and JiaXian.

- I. 在首页点击 course catalog 进入 course 页面。在 course 表增加表项。增加的表项的 TeacherID, 在 teacher 表里必须有对应的 TeacherID, 否则插入或修改时报错。

Courses Information

Insert Course

CourseID

EE236

Name

Algorithm

TeacherID

123

Cancel

Confirm

localhost:3000 says

insert failed

Error: [MySQL][ODBC 8.0(w) Driver][mysqld-8.0.21]Cannot add or update a child row: a foreign key constraint fails
(`school_database`.`course`, CONSTRAINT `course_ibfk_1` FOREIGN KEY (`teacher_id`) REFERENCES `teacher` (`teacher_id`) ON DELETE CASCADE)

OK

Insert Course

CourseID

EE236

Name

Algorithm

TeacherID

J1064

Cancel

Confirm

CourseID	Name	TeacherID		
EE236	Algorithm	J1064	delete	update

- m. 插入表项或修改表项时，若属性不是主码或外码，则属性可以为 NULL 值，否则出现错误提示。

Courses Information

Insert Course

CourseID

L121

Name

TeacherID

H123

Cancel

Confirm

CourseID	Name	TeacherID		
EE236	Algorithm	J1064	delete	update
L121		H123	delete	update

Name

TeacherID

Cancel

Confirm

localhost:3000 says
confirm update?

OKCancel

return

earch..

CourseID	Name	TeacherID		
EE236	Algorithm	J1064	delete	update
<div>L121</div>	<div>introduction to literature</div>	<div>H123</div>	<div>cancel</div>	<div>confirm</div>

CourseID	Name	TeacherID		
EE236	Algorithm	J1064	delete	update
L121	introduction to literature	H123	delete	update

- n. 由于 course 表的 teacherID 引用了 teacher 表定义的 teacherID，当 teacher 表的某表项被删除时，在 course 表引用了此项的 teacherID 的项也会被删除。

Teachers Information

Insert Teacher

return

Search..

localhost:3000 says
confirm delete?

OK

Cancel

TeacherID	Name	Department		
H123	Lee	Literature	<div>delete</div>	<div>update</div>
J1064	Wong	EE	<div>delete</div>	<div>update</div>

TeacherID	Name	Department		
J1064	Wong	EE	<div>delete</div>	<div>update</div>

CourseID	Name	TeacherID		
EE236	Algorithm	J1064	<div>delete</div>	<div>update</div>