

《数据库原理与应用》实验报告

实验名称	B/S结构的数据库编程	实验序号	3
实验日期	Fri. Nov. 4 2022	实验人	苏靖博

1、实验目的

1. 学会在IE浏览器中查看 React 网页的执行结果.
2. 掌握 React 框架实现动态网页的制作.

二、实验任务

通过设计ASP.NET网页，实现“实验一”中的1-11个语句执行结果.

项目链接 <https://github.com/Sue217/NCUT/tree/main/DataBase/labs/0x02>

SQL APPLICATION

--- Jingbo Su from NCUT ---

Insert	Task 1	Task 2	Task 3	Task 4	Task 5
Task 6	Task 7	Task 8	Task 9	Task A	Task B

Lower Bound (ONLY for task 11)

Higher Bound (ONLY for task 11)

Submit

3、 关键程序代码与执行结果截图

(1) 求全体学生的学号、姓名和出生年份。

001	Jingbo Su	2002
002	Harry Potter	2000
003	Ron Weasley	2000
004	Hermione Granger	2001
005	Draco Malfoy	2000
006	Tom Riddle	1926

- client

```

<button className='opt_btn' onClick={() => { setButton(1)
}}>Task 1</button>
case 1:
    return <Table1 key={val.sname} val={val} />;

```

- server

```

app.get('/api/get/task1', (req, res) => {
    const sqlSelect = tasks(1);
    db.query(sqlSelect, (err, result) => {
        res.send(result);
    });
});

```

(2) 求每个系的学生总人数并按降序排列。

Gryffindor	3
Slughorn	2
Computer Science	1

- client

```

<button className='opt_btn' onClick={() => { setButton(2)
}}>Task 1</button>
case 2:

```

```
return <Table2 key={val.sdept} val={val} />;
```

- **server**

```
app.get('/api/get/task2', (req, res) => {
  const sqlSelect = tasks(2);
  db.query(sqlSelect, (err, result) => {
    res.send(result);
  });
});
```

- (3) 求选修了课程号为**002**或**003**的学生的学号、课程名和成绩。

001	Advanced Programming	100
002	Advanced Programming	100
003	Advanced Programming	96
004	Advanced Programming	100
004	Operation System	100
005	Operation System	93
006	Advanced Programming	100
006	Operation System	100

- **client**

```
<button className='opt_btn' onClick={() => { setButton(3)
}}>Task 1</button>
case 3:
  return <Table3 key={[val.cname][val.sno]} val={val} />;
```

- **server**

```
app.get('/api/get/task3', (req, res) => {
  const sqlSelect = tasks(3);
  db.query(sqlSelect, (err, result) => {
    res.send(result);
  });
});
```

- (4) 检索选修某课程的学生人数多于**3**人的教师姓名。

Albus Dumbledore
Remus Lupin
Rubeus Hagrid
Minerva McGonagall
Olivia Green

- **client**

```
<button className='opt_btn' onClick={() => { setButton(4)
  }}>Task 1</button>
case 4:
  return <Table4 key={val.tname} val={val} />;
```

- **server**

```
app.get('/api/get/task4', (req, res) => {
  const sqlSelect = tasks(4);
  db.query(sqlSelect, (err, result) => {
    res.send(result);
  });
});
```

(5) 查询所有未选课程的学生姓名和所在系。

— NULL —

- **client**

```
<button className='opt_btn' onClick={() => { setButton(5)
  }}>Task 1</button>
case 5:
  return <Table5 key={val.sname} val={val} />;
```

- **server**

```
app.get('/api/get/task5', (req, res) => {
  const sqlSelect = tasks(5);
  db.query(sqlSelect, (err, result) => {
    res.send(result);
  });
});
```

(6) 求每个同学的课程成绩的最高分，查询结果项包括学生姓名、课程号及最高分。

Jingbo Su	050	100
Jingbo Su	193	100
Jingbo Su	221	100
Jingbo Su	61b	100
Harry Potter	050	100
Harry Potter	221	100
Harry Potter	61b	100
Ron Weasley	75p	100
Hermione Granger	050	100
Hermione Granger	193	100
Hermione Granger	221	100
Hermione Granger	61b	100
Hermione Granger	75p	100
Hermione Granger	828	100
Draco Malfoy	050	100
Tom Riddle	050	100
Tom Riddle	193	100
Tom Riddle	221	100
Tom Riddle	61b	100
Tom Riddle	75p	100
Tom Riddle	828	100

- **client**

```

<button className='opt_btn' onClick={() => { setButton(6)
}}>Task 1</button>

case 6:
    return <Table6 key={[val.cno][val.sname]} val={val} />;

```

- **server**

```
app.get('/api/get/task6', (req, res) => {
  const sqlSelect = tasks(6);
  db.query(sqlSelect, (err, result) => {
    res.send(result);
  });
});
```

(7) 求所有讲授数据结构课程的教师姓名。

Olivia Green

- client

```
<button className='opt_btn' onClick={() => { setButton(7)
}}>Task 1</button>
case 7:
  return <Table7 key={val.tname} val={val} />;
```

- server

```
app.get('/api/get/task7', (req, res) => {
  const sqlSelect = tasks(7);
  db.query(sqlSelect, (err, result) => {
    res.send(result);
  });
});
```

(8) 查询所有选修了 **Severus Snape** 老师的课程的学生信息。

004	Hermione Granger	Female	21	Gryffindor
005	Draco Malfoy	Male	22	Slughorn
006	Tom Riddle	Male	96	Slughorn

- client

```
<button className='opt_btn' onClick={() => { setButton(8)
}}>Task 1</button>
```

```
case 8:
  return <Table8 key={[val.sname][val.ssex]} val={val} />;
```

- **server**

```
app.get('/api/get/task8', (req, res) => {
  const sqlSelect = tasks(8);
  db.query(sqlSelect, (err, result) => {
    res.send(result);
  });
});
```

- (9) 新建一个关系表**S1(SNO,SNAME,SD,SA)**, 其字段类型定义与**Student**表中的相应字段(**SNO,SNAME,SDEPT,SAGE**)的数据类型定义相同。将表**Student**中在计算机系‘**CS**’的学生数据插入到表**S1**中。

001	Jingbo Su	Computer Science	20
-----	-----------	------------------	----

- **client**

```
<button className='opt_btn' onClick={() => { setButton(9)
}}>Task 1</button>
```

```
case 9:
```

```
  return <Table9 key={[val.sd][val.sno]} val={val} />;
```

- **server**

```
app.get('/api/get/task9', (req, res) => {
  const sqlSelect = tasks(9);
  db.query(sqlSelect, (err, result) => {
    res.send(result);
  });
});
```

- (10) 建立计算机系的学生的视图**STUDENT_CS**。利用视图**STUDENT_CS**, 将学生的**001**号课程的成绩提高**5**分。

001	Jingbo Su	100
001	Jingbo Su	100
001	Jingbo Su	100
001	Jingbo Su	100

- **client**

```
<button className='opt_btn' onClick={() => { setButton(10)
}}>Task 1</button>
case 10:
  return <TableA key={[val.sno][val.sname]} val={val} />;
```

- **server**

```
app.get('/api/get/task10', (req, res) => {
  const sqlSelect = tasks(10);
  db.query(sqlSelect, (err, result) => {
    res.send(result);
  });
});
```

- (11) 创建一个带参数的存储过程，将数据结构课程的成绩在**low**与**Hign**分数段的学生信息提取出来，查询后的结果包括：姓名、所在的系、成绩。

SQL APPLICATION

--- Jingbo Su from NCUT ---

Insert

Task 1

Task 2

Task 3

Task 4

Task 5

Task 6

Task 7

Task 8

Task 9

Task A

Task B

Lower Bound (ONLY for task 11)

85

Higher Bound (ONLY for task 11)

100

Submit

Harry Potter	Gryffindor	90
Ron Weasley	Gryffindor	100
Hermione Granger	Gryffindor	100
Tom Riddle	Slughorn	100

- client

```
<button className='opt_btn' onClick={() => { setButton(11)
}}>Task B</button>
case 11:
  return <TableB key={[val.sname][val.sdept]} val={val} />;
```

- server

```
var lower = 0;
var higher = 0;

app.post('/api/fetch/values', (req, res) => {
  lower = parseInt(req.body.min);
  higher = parseInt(req.body.max);
```

```
});  
  
app.get('/api/fetch/result', (req, res) => {  
  // console.log(lower, higher, typeof(lower), typeof(higher));  
  const sqlCall = "CALL __PROCEDURE_TASK__(?,?)";  
  db.query(sqlCall, [lower, higher], (err, result) => {  
    res.send(result);  
  })  
});
```

4、 实验中的问题及解决方法

实验中最大的麻烦是学习全新的知识, 包括 JavaScript 语句和基本的 React.js 框架, 在经历的大概半星期的学习基本掌握了在该框架下编程、实现数据库的连接及逻辑实现;

技术上遇到的问题是页面在前端无数次 re-render , 是在实验3中发现的这个问题, 解决及修改方案也比较简单, 即尝试修改按钮每处单独写 onClick , 即不让 button 事件持续通过刷新等待事件发生, 这样就能够在事件触发的时候进行反应及操作, 避免页面无穷刷新和 render.

5、 实验的总结与收获

1. 更加深入学习了 JavaScript 及 ES6 相关语法
2. 掌握了使用 React 框架及连接数据库编程
3. 掌握了 C/S 架构应用的实现方式
4. 掌握了在浏览器中调试网页相关操作