INFS7901 Completed Project

Student information management and school educational administration application

Aim of the project :

The most important content of this application is the student information, so all the functions of this applications are carried out around these data information. Therefore, this application make it convenient for for teachers and departments to query and update or delete some information about students.

Queries:

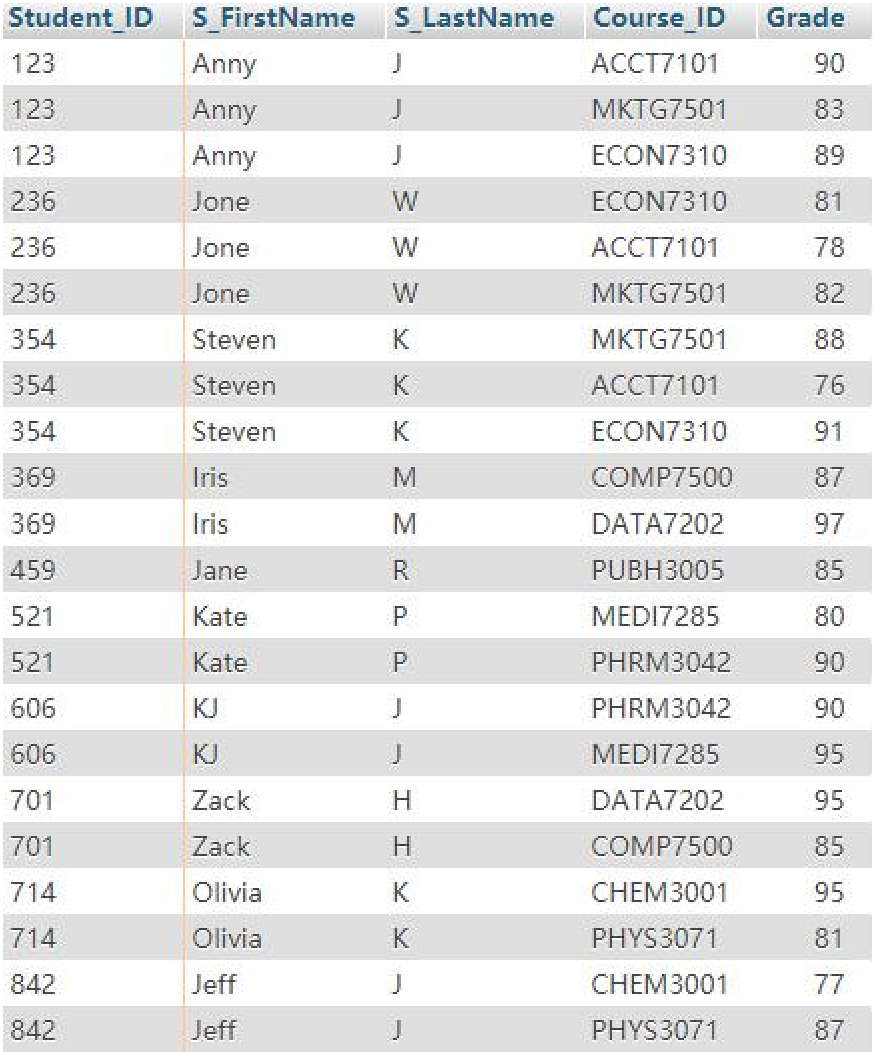
(1)
Type of SQL query:
join query

SQL statement:

```
SELECT S.`Student_ID`,`S_FirstName`,`S_LastName`,`Course_ID`,Grade
FROM Student S,Enrollment E
WHERE S.`Student_ID`= E.`Student_ID`
```

Function:
This statement is implemented to provide the list of the grade of each course of each student.

Pictorial UI:



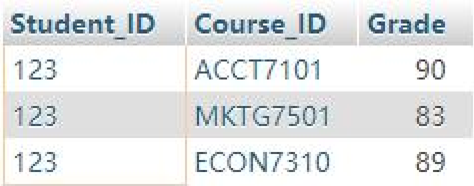
Student_ID	S_FirstName	S_LastName	Course_ID	Grade
123	Anny	J	ACCT7101	90
123	Anny	J	MKTG7501	83
123	Anny	J	ECON7310	89
236	Jone	W	ECON7310	81
236	Jone	W	ACCT7101	78
236	Jone	W	MKTG7501	82
354	Steven	K	MKTG7501	88
354	Steven	K	ACCT7101	76
354	Steven	K	ECON7310	91
369	Iris	M	COMP7500	87
369	Iris	M	DATA7202	97
459	Jane	R	PUBH3005	85
521	Kate	P	MEDI7285	80
521	Kate	P	PHRM3042	90
606	KJ	J	PHRM3042	90
606	KJ	J	MEDI7285	95
701	Zack	H	DATA7202	95
701	Zack	H	COMP7500	85
714	Olivia	K	CHEM3001	95
714	Olivia	K	PHYS3071	81
842	Jeff	J	CHEM3001	77
842	Jeff	J	PHYS3071	87

(2)
Type of SQL query:
division query

Function:
This statement is implemented to provide the list of the grade of each course of the specific student.

①
SQL statement:
SELECT `Student_ID`,`Course_ID`,`Grade`
FROM `Enrollment`
WHERE `Student_ID`=123

Pictorial UI:



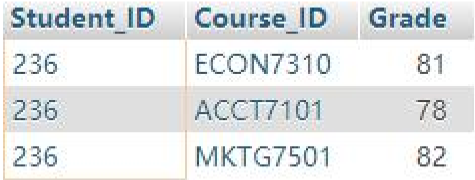
Student_ID	Course_ID	Grade
123	ACCT7101	90
123	MKTG7501	83
123	ECON7310	89

②

SQL statement:

```
SELECT `Student_ID`,`Course_ID`,`Grade`
FROM `Enrollment`
WHERE `Student_ID`=236
```

Pictorial UI:



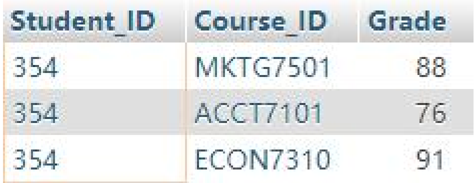
Student_ID	Course_ID	Grade
236	ECON7310	81
236	ACCT7101	78
236	MKTG7501	82

③

SQL statement:

```
SELECT `Student_ID`,`Course_ID`,`Grade`
FROM `Enrollment`
WHERE `Student_ID`=354
```

Pictorial UI:



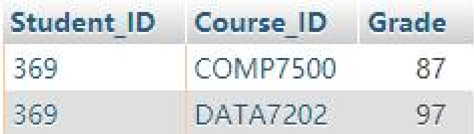
Student_ID	Course_ID	Grade
354	MKTG7501	88
354	ACCT7101	76
354	ECON7310	91

④

SQL statement:

```
SELECT `Student_ID`,`Course_ID`,`Grade`
FROM `Enrollment`
WHERE `Student_ID`=369
```

Pictorial UI:



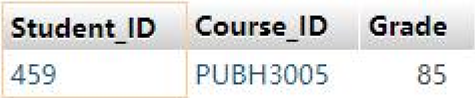
Student_ID	Course_ID	Grade
369	COMP7500	87
369	DATA7202	97

⑤

SQL statement:

```
SELECT `Student_ID`,`Course_ID`,`Grade`
FROM `Enrollment`
WHERE `Student_ID`=459
```

Pictorial UI:



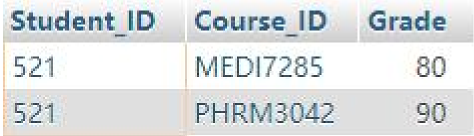
Student_ID	Course_ID	Grade
459	PUBH3005	85

⑥

SQL statement:

```
SELECT `Student_ID`,`Course_ID`,`Grade`
FROM `Enrollment`
WHERE `Student_ID`=521
```

Pictorial UI:



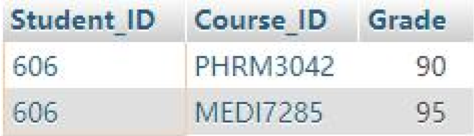
Student_ID	Course_ID	Grade
521	MEDI7285	80
521	PHRM3042	90

⑦

SQL statement:

```
SELECT `Student_ID`,`Course_ID`,`Grade`
FROM `Enrollment`
WHERE `Student_ID`=606
```

Pictorial UI:



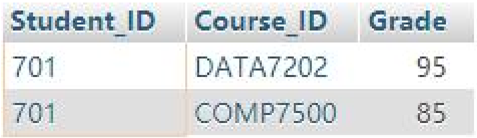
Student_ID	Course_ID	Grade
606	PHRM3042	90
606	MEDI7285	95

⑧

SQL statement:

```
SELECT `Student_ID`,`Course_ID`,`Grade`
FROM `Enrollment`
WHERE `Student_ID`=701
```

Pictorial UI:



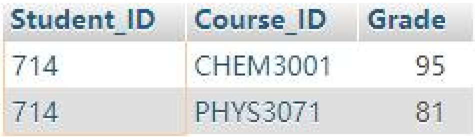
Student_ID	Course_ID	Grade
701	DATA7202	95
701	COMP7500	85

⑨

SQL statement:

```
SELECT `Student_ID`,`Course_ID`,`Grade`
FROM `Enrollment`
WHERE `Student_ID`=714
```

Pictorial UI:



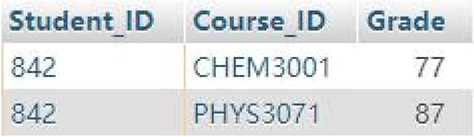
Student_ID	Course_ID	Grade
714	CHEM3001	95
714	PHYS3071	81

⑩

SQL statement:

```
SELECT `Student_ID`,`Course_ID`,`Grade`
FROM `Enrollment`
WHERE `Student_ID`=842
```

Pictorial UI:



Student_ID	Course_ID	Grade
842	CHEM3001	77
842	PHYS3071	87

(3)

Type of SQL query:

aggregation query

① max function

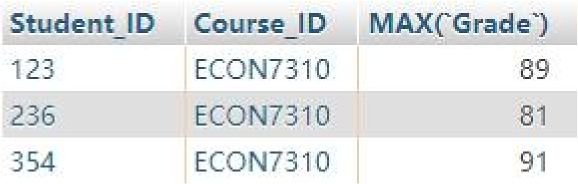
SQL statement:

```
SELECT `Student_ID`,`Course_ID`,MAX(`Grade`)
FROM Enrollment
WHERE `Course_ID`='ECON7310'
GROUP BY`Student_ID`,`Course_ID`
```

Function:

This statement is implemented to provide the list of the grade of students who take ECON7310, which is easily for teacher to find out whose grade of this course is the highest.

Pictorial UI:



Student_ID	Course_ID	MAX(`Grade`)
123	ECON7310	89
236	ECON7310	81
354	ECON7310	91

②avg function

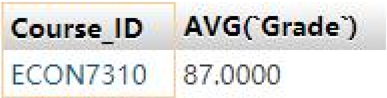
SQL statement:

SELECT `Course_ID`,AVG(`Grade`)
FROM Enrollment
WHERE `Course_ID`='ECON7310'

Function:

This statement is implemented to provide the list of average grade of ECON7310.

Pictorial UI:



Course_ID	AVG(`Grade`)
ECON7310	87.0000

(4)

Type of SQL query:

Update operation

SQL statement:

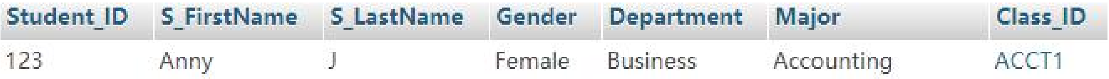
UPDATE Student
SET `Department`='Engineering', `Major` = 'Data Science', `Class_ID`= 'DS1'
WHERE `Student_ID` = 123

Function:

This statement is implemented to update the information of student whose Student_ID=123 when this student change her major.

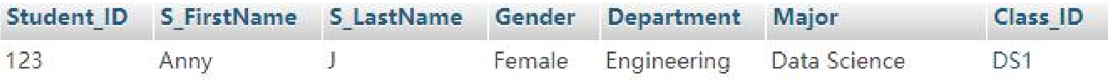
Pictorial UI:

Before updating:



Student_ID	S_FirstName	S_LastName	Gender	Department	Major	Class_ID
123	Anny	J	Female	Business	Accounting	ACCT1

After updating:



Student_ID	S_FirstName	S_LastName	Gender	Department	Major	Class_ID
123	Anny	J	Female	Engineering	Data Science	DS1

GUI code

(1) Create Student class（Student.php）

```
<?php

class Student
{
        private $Student_ID;
        private $S_FirstName;
        private $S_LastName;
        private $Gender;
        private $Department;
        private $Major;
        private $Class_ID;


        /**
          * @return mixed
          */
        public function getStudent_ID()
        {
                return $this->Student_ID;
        }

        /**
          * @param mixed $Student_ID
          */
        public function setStudent_ID($Student_ID)
        {
                $this->Student_ID = $Student_ID;
        }

        /**
          * @return mixed
          */
        public function getS_FirstName()
        {
                return $this->S_FirstName;
        }

        /**
          * @param mixed $S_FirstName
          */
```

```
public function setS_FirstNameName($S_FirstName)
{
        $this->S_FirstName = $S_FirstName;
}

/**
  * @return mixed
  */
public function getS_LastName()
{
        return $this->S_LastName;
}

/**
  * @param mixed $S_LastName
  */
public function setS_LastName($S_LastName)
{
        $this->S_LastName = $S_LastName;
}

/**
  * @return mixed
  */
public function getGender()
{
        return $this->Gender;
}

/**
  * @param mixed $Gender
  */
public function setGender($Gender)
{
        $this->gender = $Gender;
}

/**
  * @return mixed
  */
public function getDepartment()
{
        return $this->Department;
}
```

```
/**
  * @param mixed $Department
  */
public function setDepartment($Department)
{
        $this->Department = $Department;
}

/**
  * @return mixed
  */
public function getMajor()
{
        return $this->Major;
}

/**
  * @param mixed $Major
  */
public function setMajor($Major)
{
        $this->telephone = $Major;
}

/**
  * @return mixed
  */
public function getClass_ID()
{
        return $this->Class_ID;
}

/**
  * @param mixed $Class_ID
  */
public function setClass_ID($Class_ID)
{
        $this->Student_ID = $Class_ID;
}


function __toString()
{
```

```
            return                                  "Student{Student_ID=$this->Student_ID,
S_FirstName='$this->S_FirstName', S_LastName='$this->S_LastName',
            Gender='$this->Gender',                    Department='$this->Department',
Major='$this->Major',
            Class_ID='$this->Class_ID'}";
      }
}
?>
```

(2) Connect to database（DataSource.php）

```
<?php
      function getConn() {
           $host = "localhost";
           $user = "root";
           $password = "root";
           $database = "student";

           $conn = mysqli_connect($host, $user, $password, $database);

           if ($conn->error) {
                 die("Connect database[$database]fail！ ");
           }
           return $conn;
      }
?>
```

(3) Access students' data（StudentDao.php）[including GUI using select, update, delete, insert query]

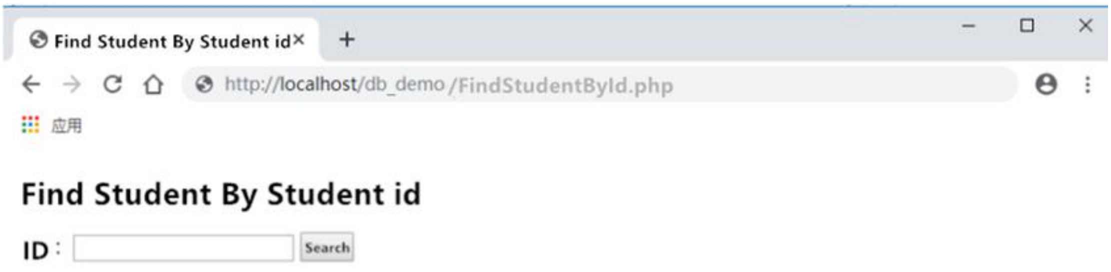
```
<?php
require("Student.php");
require("DataSource.php");

/**
  * query all students
  *
  * @return array
  */
function findAllStudents() {
      $conn = getConn();
      $sql = "SELECT * FROM student";
      $result = $conn->query($sql);
      $students = array();
```

```
        $i = 0;

        while($row = $result->fetch_assoc()) {
                $students[$i] = new Student();
                $students[$i]->setStudent_ID($row['Student_ID']);
                $students[$i]->setS_FirstName($row['S_FirstName']);
                $students[$i]->setS_LastName($row['S_LastName']);
                $students[$i]->setGender($row['Gender']);
                $students[$i]->setDepartment($row['Department']);
                $students[$i]->setMajor($row['Major']);
                $students[$i]->setClass_ID($row['Class_ID']);
                $i++;
        }
        return $students;
}

/**
  * Query students' information by student id
  *
  * @param $Student_ID
  * @return null|Student
  */
function findStudentById($Student_ID)
{
        $conn = getConn();
        $sql = "SELECT * FROM student WHERE Student_ID = $Student_ID";
        $result = $conn->query($sql);
        if ($result->num_rows > 0) {
                $row = $result->fetch_assoc();
                // create students
                $student = new Student();
                // set attributes of students
                $student->setStudent_ID($row['Student_ID']);
                $student->setS_FirstName($row['S_FirstName']);
                $student->setS_LastName($row['S_LastName']);
                $student->setGender($row['Gender']);
                $student->setDepartment($row['Department']);
                $student->setMajor($row['Major']);
                $student->setClass_ID($row['Class_ID']);
        } else {
                $student = null;
        }
        return $student;
}
```

```
/**
  * Insert student information
  * @param $student
  * @return bool|mysqli_result
  */
function insertStudent($student)
{
	$Student_ID = $student->getStudent_ID();
	$S_FirstName = $student->getS_FirstName();
	$S_LastName = $student->getS_LastName();
	$Gender = $student->getGender();
	$Department = $student->getDepartment();
	$Major = $student->getMajor();
	$Class_ID = $student->getClass_ID();

	$conn = getConn();
	$sql = "INSERT INTO student VALUES ('$Student_ID', '$S_FirstName',
'$S_LastName', '$Gender', '$Department','$Major','$Class_ID')";
	// implement insert operation
	$retval = $conn->query($sql);

	return $retval;
}

/**
  * Update student information
  * @param $student
  * @return bool|mysqli_result
  */
function updateStudent($student)
{
	$Student_ID = $student->getStudent_ID();
	$S_FirstName = $student->getS_FirstName();
	$S_LastName = $student->getS_LastName();
	$Gender = $student->getGender();
	$Department = $student->getDepartment();
	$Major = $student->getMajor();
	$Class_ID = $student->getClass_ID();

	$conn = getConn();
	$sql = "UPDATE student SET
S_FirstName='{$S_FirstName}',S_LastName='{$S_LastName}',Gender='{$Gender}',De
partment='{$Department}',Major='{$Major}',Class_ID='{$Class_ID}' WHERE
Student_ID='{$Student_ID}'";
        // implement update operation
        $retval = $conn->query($sql);

        return $retval;
}

/**
  * Delete students' information by student id
  *
  * @param $Student_ID
  */
function deleteStudentById($Student_ID) {
        $conn = getConn();
        $sql = "DELETE FROM student WHERE Student_ID = $Student_ID";
        // implement delete operation
        $retval = $conn->query($sql);

        return $retval;
}
?>
```

(4) Find student information by student id（FindStudentById.php）

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <title>Find Student By Student id</title>
</head>
<body>
<h3>Find Student By Student id</h3>
<form action="doFindStudentById.php" method="post">
        Student_ID：<input type="text" name="ID">
        <input type="submit" value="query"><br>
</form>
</body>
</html>
```



Find Student By Student id
http://localhost/db_demo/FindStudentById.php
应用

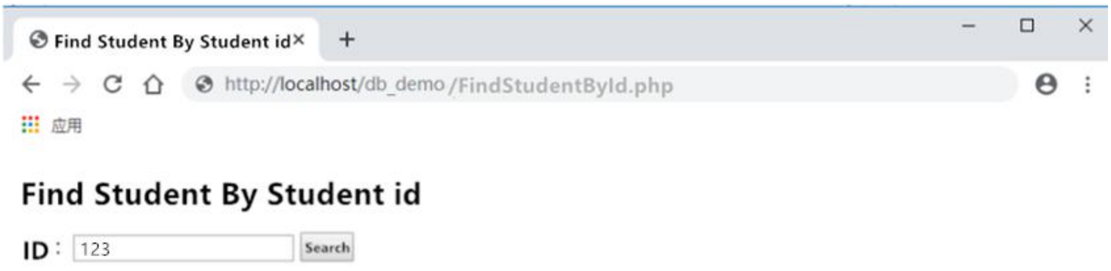
Find Student By Student id

ID：

(5) Handle page of finding student's data by student id（doFindStudentById.php）

```
<?php
	require("StudentDao.php");

	$Student_ID = $_POST['Student_ID'];
	$student = findStudentById($Student_ID);
	if ($student == null) {
		echo "The student you are looking for does not exist!";
	} else {
		echo $student;
	}
?>
```

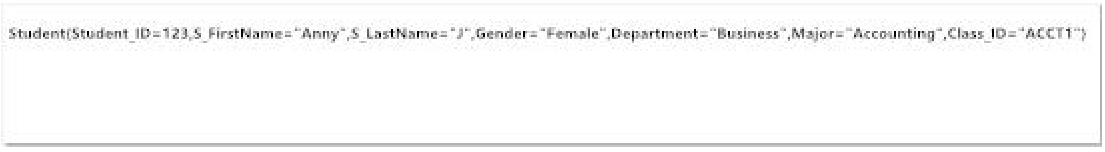
① Looking for existing student id



Find Student By Student id
http://localhost/db_demo/FindStudentById.php
应用

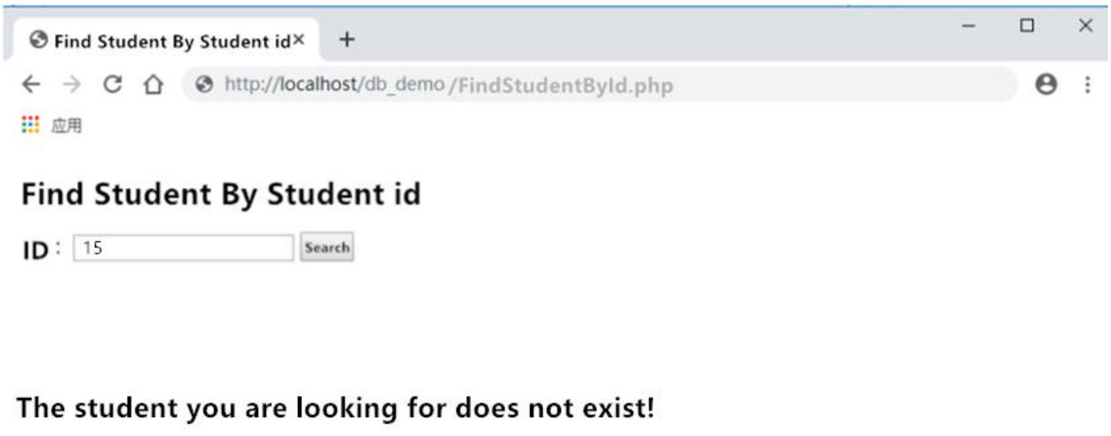
Find Student By Student id

ID： 123



```
Student(Student_ID=123,S_FirstName="Anny",S_LastName="J",Gender="Female",Department="Business",Major="Accounting",Class_ID="ACCT1")
```

② Looking for student id which is not exist


Find Student By Student id
http://localhost/db_demo/FindStudentById.php
应用

Find Student By Student id

ID： 15 Search

The student you are looking for does not exist!

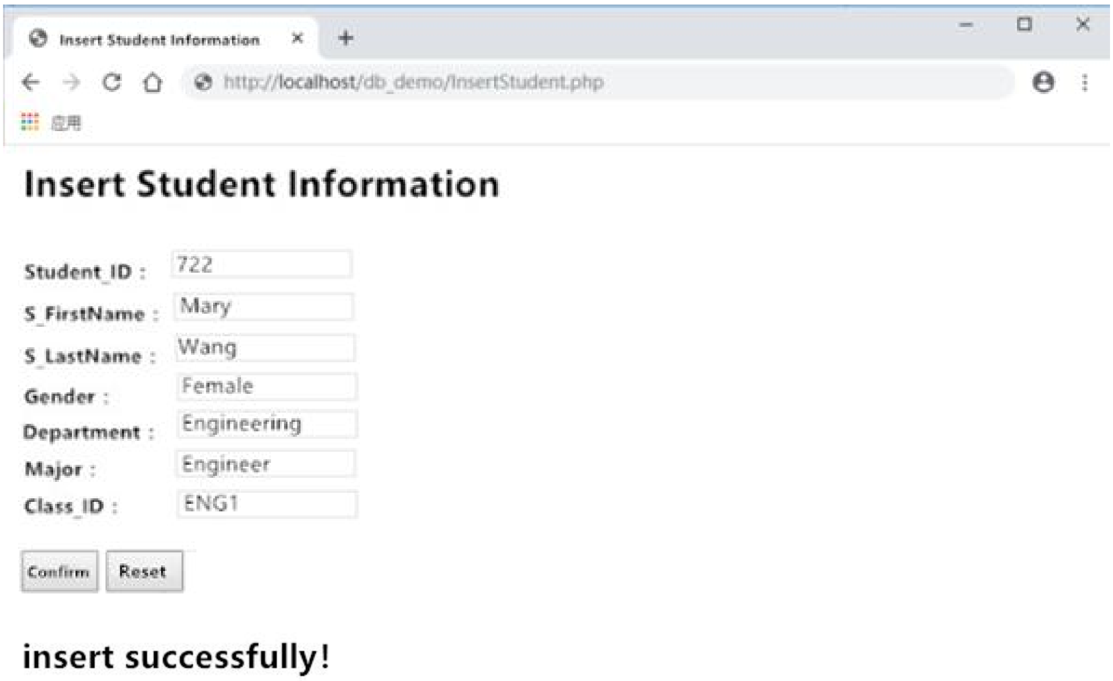
(6) Insert the student’s data（InsertStudent.php）

```
<!DOCTYPE html>
<html lang="en">
<head>
	<meta charset="UTF-8">
	<title>Insert Student Information</title>
</head>
<body>
<h3>Insert Student Information</h3>
<form action="doInsertStudent.php" method="post">
	Student_ID：<input type="number" name="Student_ID"><br>
	S_FirstName：<input type="text" name="S_FirstName"><br>
	S_LastName：<input type="text" name="S_LastName"><br>
	Gender：<input type="text" name="Gender"><br>
	Department：<input type="text" name="Department"><br>
	Major：<input type="text" name="Major"><br>
	Class_ID：<input type="number" name="Class_ID"><br>
	<input type="submit" value="confirm">
	<input type="reset" value="reset">
</form>
</body>
</html>
```

(7) Handle page of inserting student’s data（doInsertStudent.php）

```
<?php
	require("StudentDao.php");

	$Student_ID = $_POST['Student_ID'];
	$S_FirstName = $_POST['S_FirstName'];
	$S_LastName = $_POST['S_LastName'];
```

```
        $Gender = $_POST['Gender'];
        $Department = $_POST['Department'];
        $Major = $_POST['Major'];
        $Class_ID = $_POST['Class_ID'];

        $student = new Student();

        $student->setStudent_ID($Student_ID);
        $student->setS_FirstName($S_FirstName);
        $student->setS_LastName($S_LastName);
        $student->setGender($Gender);
        $student->setDepartment($Department);
        $student->setMajor($Major);
        $student->setClass_ID($Class_ID);

        $retval = insertStudent($student);

        if ($retval) {
                echo "insert successfully！";
        } else {
                echo "insert failed！";
        }
?>
```


Insert Student Information
http://localhost/db_demo/InsertStudent.php
应用

Insert Student Information

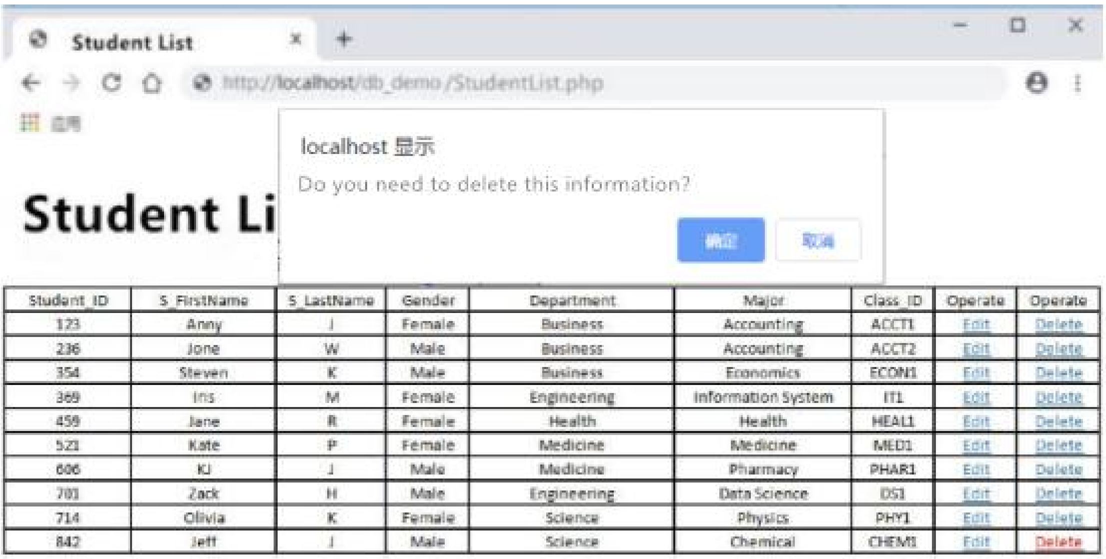
Student_ID：	722
S_FirstName：	Mary
S_LastName：	Wang
Gender：	Female
Department：	Engineering
Major：	Engineer
Class_ID：	ENG1

insert successfully!

(8) Delete student’s information（deleteStudentById.php）

```
<?php
	require("StudentDao.php");

	$id = $_GET['Student_ID'];
	$retval = deleteStudentById($id);
	$location="StudentList.php?retval=$retval";
	header("location:$location");
?>
```

① Delete student’s information whose Student_ID=842



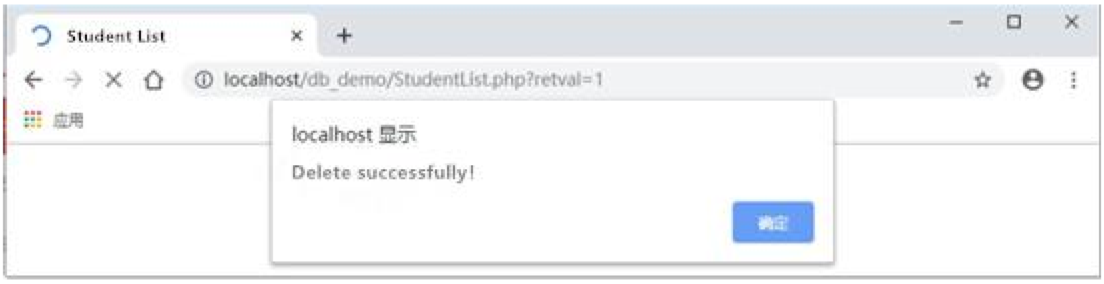
Student Li

localhost 显示

Do you need to delete this information?

Student_ID	S_FirstName	S_LastName	Gender	Department	Major	Class_ID	Operate	Operate
123	Anny	J	Female	Business	Accounting	ACCT1	Edit	Delete
236	Jone	W	Male	Business	Accounting	ACCT2	Edit	Delete
354	Steven	K	Male	Business	Economics	ECON1	Edit	Delete
369	Iris	M	Female	Engineering	Information System	IT1	Edit	Delete
459	Jane	R	Female	Health	Health	HEAL1	Edit	Delete
521	Kate	P	Female	Medicine	Medicine	MED1	Edit	Delete
606	KJ	J	Male	Medicine	Pharmacy	PHAR1	Edit	Delete
701	Zack	H	Male	Engineering	Data Science	DS1	Edit	Delete
714	Olivia	K	Female	Science	Physics	PHY1	Edit	Delete
842	Jeff	J	Male	Science	Chemical	CHEM1	Edit	Delete

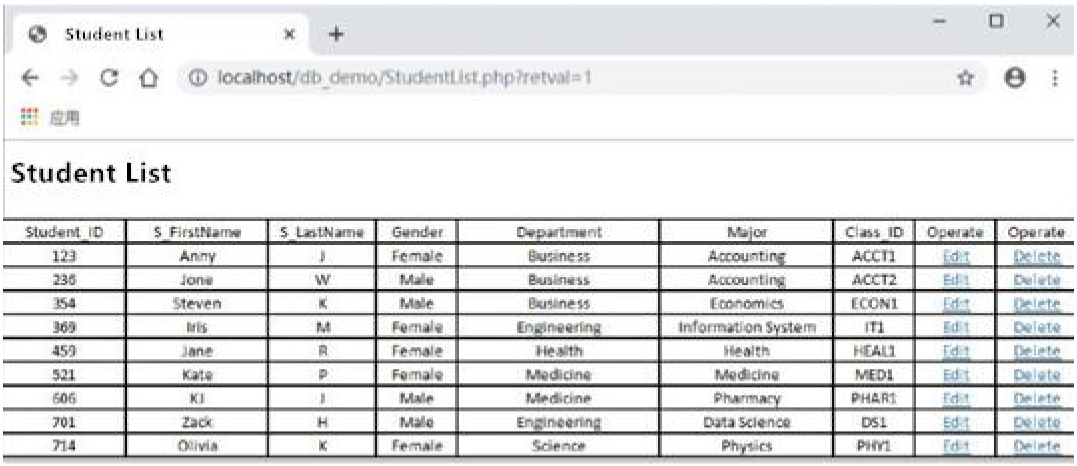
② Delete successfully



localhost 显示

Delete successfully!

③ Checking student list, that student’s information has been deleted.

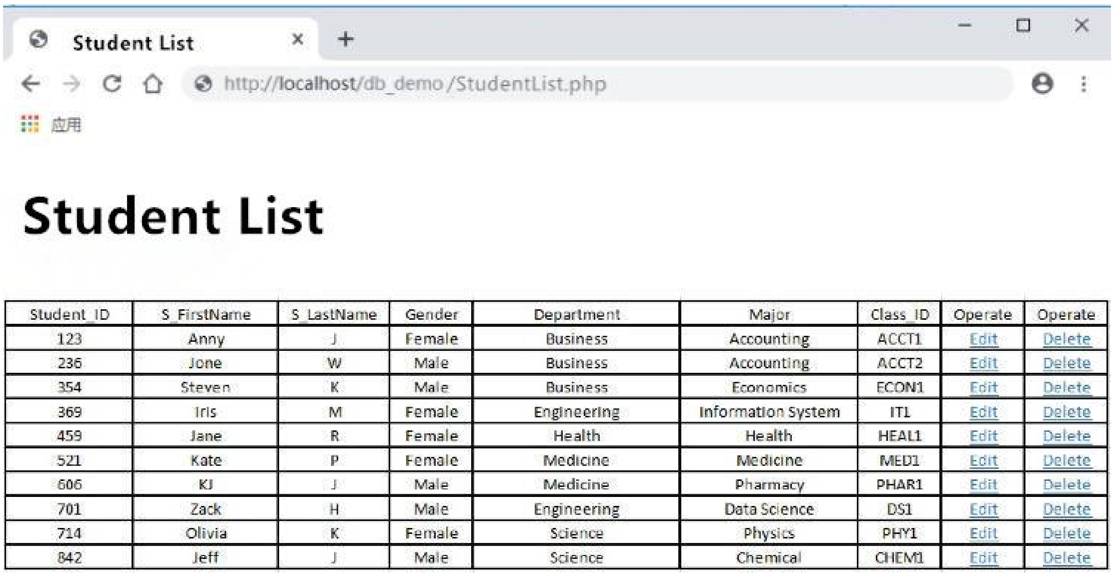


Student List

Student_ID	S_FirstName	S_LastName	Gender	Department	Major	Class_ID	Operate	Operate
123	Anny	J	Female	Business	Accounting	ACCT1	Edit	Delete
236	Jone	W	Male	Business	Accounting	ACCT2	Edit	Delete
354	Steven	K	Male	Business	Economics	ECON1	Edit	Delete
369	Iris	M	Female	Engineering	Information System	IT1	Edit	Delete
450	Jane	R	Female	Health	Health	HEAL1	Edit	Delete
521	Kate	P	Female	Medicine	Medicine	MED1	Edit	Delete
606	KJ	J	Male	Medicine	Pharmacy	PHAR1	Edit	Delete
701	Zack	H	Male	Engineering	Data Science	DS1	Edit	Delete
714	Olivia	K	Female	Science	Physics	PHY1	Edit	Delete

(9) Present page of student list（StudentList.php）

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Student List</title>
</head>
<body>
<h3>Student List</h3>
<?php
    require("StudentDao.php");

    @$retval = $_GET['retval'];
    if ($retval) {
        echo "<script type='text/javascript'>alert('Delete successfully！');</script>";
    }

    $students = findAllStudents();
    if (count($students) > 0) {
        echo "<table border='1' cellpadding='5' cellspacing='0'>";
        echo
"<tr><td>Student_ID</td><td>S_FirstName</td><td>S_LastName</td><td>Gender<
/td>"
            ."<td>Department</td><td>Major</td><td>Class_ID</td><td>Operat
e</td><td>Operate</td></tr>";
        foreach ($students as $student) {
            $Student_ID = $student->getStudent_ID();
            $S_FirstName = $student->getS_FirstName();
            $S_LastName = $student->getS_LastName();
            $Gender = $student->getGender();
            $Department = $student->getDepartment();
            $Major = $student->getMajor();
            $Student_ID = $student->getStudent_ID();
            echo
"<tr><td>$Student_ID</td><td>$S_FirstName</td><td>$S_LastName</td><td>$Gen
der</td>"
                ."<td>$Department</td><td>$Major</td><td>$Class_ID</td>"
                ."<td><a href='#'>Edit</a></td>"
                ."<td><a              href='DeleteStudentById.php?id=$Student_ID'
onclick='return deleteStudent();'>Delete</a></td></tr>";
        }
        echo "</table>";
    } else {
        echo "There is no data in student list！ ";
    }
?>
<script type="text/javascript">
    function deleteStudent() {
        var choice = confirm("Do you need to delete this information？ ");
        return choice;
    }
</script>
</body>
</html>
```



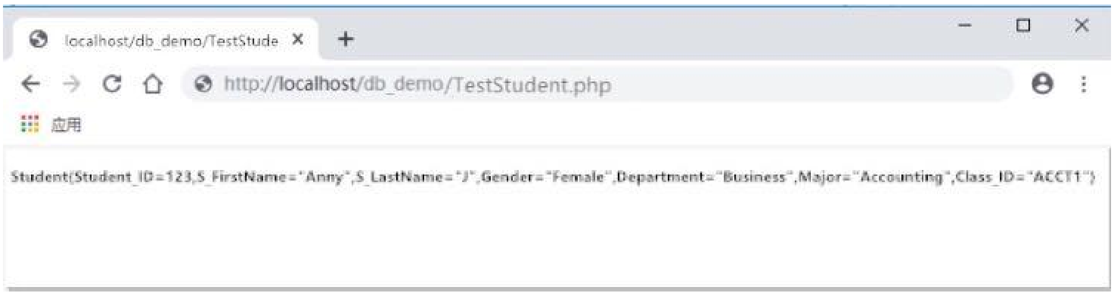
Student List

Student_ID	S_FirstName	S_LastName	Gender	Department	Major	Class_ID	Operate	Operate
123	Anny	J	Female	Business	Accounting	ACCT1	Edit	Delete
236	Jone	W	Male	Business	Accounting	ACCT2	Edit	Delete
354	Steven	K	Male	Business	Economics	ECON1	Edit	Delete
369	Iris	M	Female	Engineering	Information System	IT1	Edit	Delete
459	Jane	R	Female	Health	Health	HEAL1	Edit	Delete
521	Kate	P	Female	Medicine	Medicine	MED1	Edit	Delete
606	KJ	J	Male	Medicine	Pharmacy	PHAR1	Edit	Delete
701	Zack	H	Male	Engineering	Data Science	DS1	Edit	Delete
714	Olivia	K	Female	Science	Physics	PHY1	Edit	Delete
842	Jeff	J	Male	Science	Chemical	CHEM1	Edit	Delete

(10) Test student information page（TestStudent.php）

```
<?php
    require("Student.php");
```

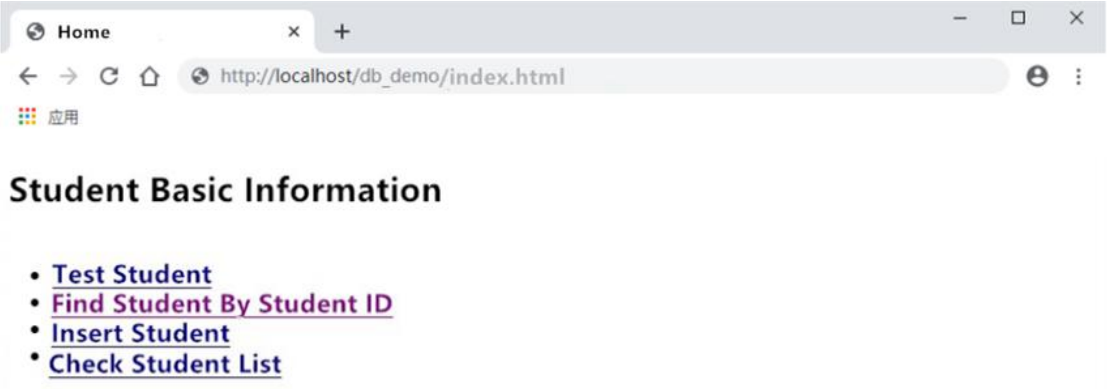
```
        $student = new Student();

        $student->setStudent_ID(123);
        $student->setS_FirstName("Anny");
        $student->setS_LastName("J");
        $student->setGender("Female");
        $student->setDepartment("Business");
        $student->setMajor("Accounting");
        $student->setClass_ID("ACCT1")

        echo $student;
?>
```


localhost/db_demo/TestStude
http://localhost/db_demo/TestStudent.php
应用

```
Student(Student_ID=123,S_FirstName="Anny",S_LastName="J",Gender="Female",Department="Business",Major="Accounting",Class_ID="ACCT1")
```

(11) Create Home page（index.html）

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <title>Home</title>
</head>
<body>
<h1>Student Basic Information</h1>
<ul>
        <li><a href="TestStudent.php">Test Student</a></li>
        <li><a href="FindStudentById.php">Find Student By Student ID</a></li>
        <li><a href="InsertStudent.php">Insert Student</a></li>
        <li><a href="StudentList.php">Check Student List</a></li>
</ul>
</body>
</html>
```



Student Basic Information

- Test Student
- Find Student By Student ID
- Insert Student
- Check Student List

Your assessment of the Project:

a. **Roughly how much time did you spend on the project?**

The whole project takes me almost a month to complete it.

b. What did you like the best and least about the project?

The best part of the project I like is formal specification, because it really help me clearly understand functional dependencies and normalization. Besides, this part also help me understand and distinguish those 4 different normal forms.

However, the least part of the project is the last one, especially build a website by myself. Although there is a guide and template in our course, I still don’t understand what’s the function of some syntax want to implement.

c. What helped you learn the best in the project?

This project help me know the whole process of building a website and learn how to build a database and how to connect database to web. It also It helped me sort out the knowledge about ER diagram, functional dependencies and normalization learned in class.

d. If you could change the way that the project was organized, what would you change?

Actually, I’m not very satisfied with the form of building a web, because I still not clearly understand implement my database in a web. I suggest that teacher could also teach some knowledge about html and flask and connection between web and database not only about database, which might help student have a clear and complete system of information system.