SQL高级

- where 条件

  查询时，不添加 where 条件, 返回数据表所有行。需要添加限定条件，只返回需要的行。

  select 字段列表 from table where 条件；

- Like 模糊匹配 % 通配符

```
-- 查找姓张的人
select * from table where name like '张%';
```

- in 语法：一次查询多个符合条件的数据

```
select  字段列表 from tb where  字段 in  (value1,value2,value3);
```

- count() 获取返回数据的总条数

```
-- 查询满足条件数据的总条数
select count(*) from table where 条件
```

- 排序

```
select * from table order by  字段名称;          默认升序
select * from table order by 字段名称 desc;      降序
```

- limit 对结果集进行截取

```
select 字段列表  from  table  limit 截取的起始索引，截取的长度
```

- 联合查询（多个表联合查询）

```
select 字段列表  from  表A  join 表B  on  A.字段=B.字段   where 条件
```

PHP操作数据库

连接数据库基本步骤

1. 连接数据库
2. 准备sql语句
3. 执行sql语句
4. 获取执行的结果并分析
5. 关闭数据库

操作数据库常用API

- `mysqli_connect(IP，用户名，密码，数据库名)` 连接数据库
- `mysqli_query($link，$sql)` 执行SQL语句
- `mysqli_error($link);` 返回错误描述
- `mysqli_close($link);` 关闭连接
- `mysqli_fetch_assoc($res);` 从结果集中取得一行作为关联数组返回
- `mysqli_num_rows($res);` 返回结果集的行数
- `mysqli_affected_rows($link);` 受影响行数

sql操作注意事项：

- 使用PHP发送SQL语句前，可以先打印SQL语句，检查语句的正确性。
- 使用变量拼接SQL语句时，字段为字符串类型，需要在变量的两侧使用单、双引号包裹。可以将所有的字段外面都使用双引号包含。

```php
// 1. 连接数据库
// mysqli_connect(ip地址，用户名，密码，数据库的名称，端口号);
// 执行结果
//    1. 连接成功，返回一个数据库连接对象
//    2. 连接失败，返回 false
// @表示错误抑制符，可以抑制错误的输出
$link = @ mysqli_connect('127.0.0.1', 'root', 'root', 'test02', 3306);
// var_dump($link);
// 如果数据库连接失败
if ( !$link ) {
  // 程序结束，die 方法，结束当前程序，输出一段语句
  die("数据库连接失败");
}
echo "数据库连接成功<br>";
// 2. 准备 sql 语句：删除一条数据
$sql = "delete from stu where id = 14";
// 3. 让数据库执行 sql 语句，并分析结果
// mysqli_query(数据库连接对象，要执行的sql语句)
// 执行成功返回 true，执行失败返回 false
if ( mysqli_query( $link, $sql ) ) {
  echo "删除成功";
}
else {
  echo "删除失败<br>";
  // mysqli_error 可以查看错误消息
  echo mysqli_error($link);
}
// 4. 关闭数据库连接（挂电话）
mysqli_close( $link );
```

非查询(增删改)和查询语句（select）的区别

通过mysqli_query()函数，来执行sql语句，操作数据库

- 执行的是非查询sql语句时，mysqli_query()执行成功返回true,失败返回false
- 而执行查询的sql语句时，mysqli_query()执行成功，返回查询数据的结果集，失败返回false **查询数据逻辑如下**

```php
// 操作步骤:
// 1. 连接数据库
// 2. 准备 sql 语句
// 3. 让数据库执行 sql 语句
// 4. 分析执行结果
// 5. 关闭数据库连接
// 1. 连接
$link = @ mysqli_connect('127.0.0.1', 'root', 'root', 'test02', 3306);
if ( !$link ) {
  // 连接失败
  die('数据库连接失败');
}
// 2. 准备 sql 语句
$sql = 'select * from stu where id;';
// 3. 执行 sql 语句，分析结果
// mysqli_query
// (1) 执行非查询语句，成功 true，失败 false
// (2) 执行查询语句，成功返回结果集，失败 false
$res = mysqli_query( $link, $sql );
if ( !$res ) {
  echo mysqli_error( $link );
  die('数据库查询失败');
}
// mysqli_fetch_assoc 查询成功，从结果集中取数据，以关联数组的形式返回
// 一次只取一条数据，如果没取到，返回 null
$arr = [];
while( $row = mysqli_fetch_assoc( $res ) ) {
  // 将值推到数组中
  $arr[] = $row;
}
echo '<pre>';
print_r($arr);
echo '</pre>';
```

数据库工具函数的封装

> 为了提高代码的复用性，把数据增删改的操作封装成一个方法

```php
// 定义常量
define( 'HOST', '127.0.0.1' );
define( 'UNAME', 'root' );
define( 'PWD', 'root' );
define( 'DB', 'test02' );
define( 'PORT', 3306 );
// 非查询语句封装
// 封装一个执行非查询语句的方法，提高代码的复用性
// 参数: $sql 要执行的 sql 语句
// 返回值: true / false
function my_exec( $sql ) {
  // 1. 连接数据库
  $link = @ mysqli_connect( HOST, UNAME, PWD, DB, PORT);

  if( !$link ) {
```

```
    echo '数据库连接失败';
    return false;
  }
  // 2. 准备 sql 语句，就是传递过来的 $sql
  // 3. 执行 sql 语句，分析结果
  if ( mysqli_query( $link, $sql ) ) {
    // 执行成功
    mysqli_close( $link ); // 关闭数据库
    return true;
  }
  else {
    // 执行失败
    mysqli_close( $link ); // 关闭数据库
    return false;
  }
}
// 查询语句的封装
// 参数: $sql 要执行的 sql 语句
// 返回值:
//    (1) 成功，返回数据(二维数组)
//    (2) 失败，返回 false
function my_query( $sql ) {
  // 1. 建立连接
  $link = @ mysqli_connect( HOST, UNAME, PWD, DB, PORT );
  if ( !$link ) {
    echo "数据库连接失败";
    return false;
  }
  // 2. 准备 sql 语句 $sql
  // 3. 执行 sql 语句，分析结果
  $res = mysqli_query( $link, $sql );  // 结果集 或者 false
  if ( !$res ) {
    echo "获取数据失败<br>";
    echo mysqli_error($link);
    mysqli_close( $link );
    return false;
  }
  // 得到结果集，将结果集的所有内容取出到数组中
  $arr = [];
  while ( $row = mysqli_fetch_assoc($res) ) {
    $arr[] = $row;
  }
  mysqli_close( $link );
  return $arr; // 返回结果数组
}
```

分页

```
1 limit分页公式
（1）limit分页公式：curPage是当前第几页；pageSize是一页多少条记录
limit  (curPage-1)*pageSize,pageSize
（2）用的地方：sql语句中
select * from student limit(curPage-1)*pageSize,pageSize;
2 总页数公式
（1）总页数公式：totalRecord是总记录数；pageSize是一页分多少条记录
int totalPageNum = (totalRecord +pageSize - 1) / pageSize;
（2）用的地方：前台UI分页插件显示分页码
（3）查询总条数：totalRecord是总记录数，SELECT COUNT(*) FROM tablename
```

学生管理系统2.0基本功能

基本功能

- 添加学生功能
- 展示学生列表功能
- 删除学生功能
- 查看学生详情
- 更新学生数据

实现思路

注册功能思路：

1. 表单设计，点击提交按钮向服务器提交表单数据
2. 在后台获取表单提交的数据，保存到数据库中
    - 先获取表单的标签的数据
    - 保存上传的图片（并保存图片存储的路径）
    - 将表单的数据和图片的路径一起保存到数据库中
3. 保存完成，跳转到列表页，查看新添加的数据

展示功能思路：

1. 先从数据库中获取数据（二维数组arr）
2. 遍历二维数组，将数组中数据渲染到页面中

删除功能思路：

1. 获取要删除数据的id
2. 根据id删除数据库中指定的数据
3. 删除完毕，返回列表页

详情展示功能

1. 获取要查看详情数据的id
2. 根据id通过联合查询，获取到需要用数据
3. 把数据显示在页面中
4. 点击返回按钮，可以返回到列表页

更新数据思路： 更新数据的思路=先渲染 再 提交

1. 获取要查看详情数据的id
2. 把对应id的数据填充到修改页面中
3. 点击修改按钮，获取表单的数据，提交给服务器
4. 在服务器更新数据
5. 更新完成后跳转到列表页