

17. 数据库的高级查询

学习要点：

1. 高级查询

本节课我们要学习查询中的几种高级查询方式，了解更多拼装 SQL 的细节。

一. 高级查询

1. 使用|(OR)或&(AND)来实现 where 条件的高级查询，where 支持多个连缀；

```
$user = Db::name('user')
    ->where('username|email', 'like', '%xiao%')
    ->where('price&uid', '>', 0)
    ->select();

//生成的 SQL
SELECT * FROM `tp_user` WHERE ( `username` LIKE '%xiao%' OR `email` LIKE '%xiao%' ) AND ( `price` > 0 AND `uid` > 0 )
```

2. 关联数组方式，可以在 where 进行多个字段进行查询；

```
$user = Db::name('user')->where([
    ['id', '>', 0],
    ['status', '=', 1],
    ['price', '>=', 80],
    ['email', 'like', '%163%']
])->select();

//生成的 SQL
SELECT * FROM `tp_user` WHERE `id` > 0 AND `status` = 1 AND `price` >= 80 AND `email` LIKE '%163%'
```

3. 根据之前的课程中，条件字符串复杂组装，比如使用 exp 了，就使用 raw() 方法；

```
$user = Db::name('user')->where([
    ['status', '=', 1],
    ['price', 'exp', Db::raw('>80')]
])->select();

//生成的 SQL
SELECT * FROM `tp_user` WHERE `status` = 1 AND ( `price` >80 )
```

4. 如果有多个 where，并且 where 条件是分离的 \$map，而 \$map 本身有多个条件；
5. 那么 \$map 条件如果需要先执行出结果，再和后续条件判断，也就是加上括号；
6. 那么，需要对这个 \$map 变量，再加上一个中括号处理优先级；

```
$map = [
    ['status', '=', 1],
```

```

        ['price', 'exp', Db::raw('>80')]
    ];
    $user = Db::name('user')
        ->where([$map])
        ->where('status', '=', '%163.com%')->select();

    //生成的 SQL
    SELECT * FROM `tp_user` WHERE ( `status` = 1 AND ( `price` >80 ) ) AND `status`
    = %163.com%

```

7. 如果，条件中有多次出现一个字段，并且需要 OR 来左右筛选，可以用 whereOr;

```

    $map1 = [
        ['username', 'like', '%小%'],
        ['email', 'like', '%163%']
    ];
    $map2 = [
        ['username', 'like', '%孙%'],
        ['email', 'like', '%.com%']
    ];
    $user = Db::name('user')
        ->whereOr([$map1, $map2])->select();

    //生成的 SQL
    SELECT * FROM `tp_user` WHERE ( `username` LIKE '%小%' AND `email` LIKE '%163%' )
    OR ( `username` LIKE '%孙%' AND `email` LIKE '%.com%' )

```

8. 闭包查询可以连缀，会自动加上括号，更清晰，如果是 OR，请用 whereOR();

```

    $user = Db::name('user')->where(function ($query) {
        $query->where('id', '>', 10);
    })->whereOr(function ($query) {
        $query->where('username', 'like', '%小%');
    })->select();

```

9. 对于比较复杂或你不知道如何拼装 SQL 条件，那么就直接使用 whereRaw()即可;

```

    $user = Db::name('user')
        ->whereRaw('(username LIKE "%小%" AND email LIKE "%163%") OR (price > 80)')
        ->select();

```

10. whereRaw()方式也支持参数绑定操作，具体如下;

```

    $user = Db::name('user')
        ->whereRaw('(username LIKE :username AND email LIKE :email)
                    OR (price > :price)',
        ['username'=> '%小%', 'email'=> '%163%', 'price'=>80])
        ->select();

```