

12. 构造器的查询表达式

学习要点：

1. **select** 查询
2. **where** 查询

本节课我们来开始学习数据库的构造器查询 **select** 和 **where** 查询；

一. select 查询

1. **select()**方法可以制定你想要的列，而不是所有列；

//设置显示的列，设置列别名

```
$users = DB::table('users')->select('username as name', 'email')->get();
```

2. **addSelect()**方法，可以在你基础的查询构造器上再增加想要显示的字段；

//给已经构建好的查询添加更多字段

```
$base = DB::table('users')->select('username as name', 'email');  
$users = $base->addSelect('gender')->get();
```

3. **DB::raw()**方法可以在 **select()**内部实现原生表达式，否则解析错误；

//结合原生 SQL 实现复杂查询

```
$users = DB::table('users')->select(DB::raw('COUNT(*) AS id, gender'))  
->groupBy('gender')  
->get();
```

4. 也可以使用 **selectRaw()**方法实现内部原生；

//或者直接使用 **selectRaw()**方法实现原生

```
$users = DB::table('users')->selectRaw('COUNT(*) AS count, gender')  
->groupBy('gender')  
->get();
```

5. 还可以通过 **havingRaw()**方法实现更精准的分组筛选；

//使用 **havingRaw** 方法实现分组筛选

```
$users = DB::table('users')->selectRaw('COUNT(*) AS count, gender')  
->groupBy('gender')  
->havingRaw('count>5')  
->get();
```

二. where 查询

1. **where()**查询，即条件查询，完整形式需要字段表达式和值三个；

//where 查询完整形式

```
$users = DB::table('users')->where('id', '=', 19)->get();
```

2. 大部分情况下，是等于用的比较多，就可以省略掉=号参数；

//where 查询完整形式

```
$users = DB::table('users')->where('id', 19)->get();
```

3. 当然，还有>、<、>=、<=、<>、like 等操作符；

```
users = DB::table('users')->where('price', '>=', 95)->get();
```

```
$users = DB::table('users')->where('username', 'like', '%小%')->get();
```

4. 如果条件较多，可以用数组来分别添加条件，具体如下：

//如果条件都是等于，查看 SQL 语句用->toSql()替换->get()

```
$users = DB::table('users')->where([
    'price'    => 90,
    'gender'   => '男'
])->get();
```

//如果条件非等于

```
$users = DB::table('users')->where([
    ['price', '>=', 90],
    ['gender', '=', '男']
])->get();
```