

## 11. 构造器的查询.分块.聚合

学习要点：

1. 构造器查询
2. 分块.聚合

本节课我们来开始学习数据库的构造器查询以及分块和聚合查询。

### 一. 构造器查询

1. `table()`方法引入相应的表，`get()`方法可以查询当前表的所有数据；

//获取全部结果

```
$users = DB::table('users')->get();
```

2. `first()`方法，可以获取到第一条数据；

//获取第一条数据

```
$users = DB::table('users')->first();
```

3. `value(字段名)`方法，可以获取到第一条数据的指定字段的值；

//获取第一条数据的 email 字段值

```
$users = DB::table('users')->value('email');
```

4. `find(id)`方法，可以获取指定 id 的一条数据；

//通过 id 获取指定一条数据

```
$users = DB::table('users')->find(20);
```

5. `pluck(字段名)`可以获取所有数据单列值的集合；

//获取单列值的集合

```
$users = DB::table('users')->pluck('username');
```

```
$users = DB::table('users')->pluck('username', 'email');
```

### 二. 分块.聚合

1. 如果你一次性处理成千上万条记录，防止读取出错，可以使用 `chunk()`方法；

//切割分块执行，每次读取 3 条，id 排序；

```
DB::table('users')->orderBy('id')->chunk(3, function ($users) {  
    foreach ($users as $user) {  
        echo $user->username;  
    }  
    echo '-----<br>';  
});
```

2. 构造器查询提供了：`count()`、`max()`、`min()`、`avg()`和`sum()`聚合查询；

//聚合查询

```
return DB::table('users')->count();  
return DB::table('users')->max('price');  
return DB::table('users')->avg('price');
```

3. 构造器查询两个判断记录是否存在的方法：`exists()`和`doesntexists()`方法；

//判断是否存在

```
return DB::table('users')->where('id', 19)->exists();  
return DB::table('users')->where('id', 18)->doesntExist();
```

PS：这里 `DB::` 第一个使用静态，返回查询对象，然后使用 `->where` 等各种查询方法，这些查询方法返回的还是查询对象，所以可以继续连缀操作。最后当遇到比如 `get()` 返回结果等方法时，停止连缀。所以，返回结果必须放在最后。