

73. 一对多关联查询

学习要点：

1. hasMany 模式

本节课我们来了解关联模型中，一对多关联查询的使用方法。

一. hasMany 模式

1. hasMany 模式，适合主表关联附表，实现一对多查询，具体设置方式如下：

```
hasMany('关联模型', ['外键', '主键']);  
return $this->hasMany('Profile', 'user_id', 'id');
```

关联模型（必须）：模型名或者模型类名

外键：关联模型外键，默认的外键名规则是当前模型名+_id

主键：当前模型主键，一般会自动获取也可以指定传入

2. 在上一节课，我们了解了表与表关联后，实现的查询方案：

```
$user = UserModel::get(19);  
return json($user->profile);
```

3. 使用->profile()方法模式，可以进一步进行数据的筛选：

```
return json($user->profile()->where('id', '>', 10)->select());
```

4. 使用 has()方法，查询关联附表的主表内容，比如大于等于 2 条的主表记录：

```
UserModel::has('profile', '>=', 2)->select();
```

5. 使用 hasWhere()方法，查询关联附表筛选后记录，比如兴趣审核通过的主表记录：

```
UserModel::hasWhere('profile', ['status'=>1])->select();
```

6. 使用 save()和 saveAll()进行关联新增和批量关联新增，方法如下：

```
$user = UserModel::get(19);  
$user->profile()->save(['hobby'=>'测试喜好', 'status'=>1]);  
$user->profile()->saveAll([  
    ['hobby'=>'测试喜好', 'status'=>1],  
    ['hobby'=>'测试喜好', 'status'=>1]  
]);
```

7. 使用 together()方法，可以删除主表内容时，将附表关联的内容全部删除：

```
$user = UserModel::get(227, 'profile');  
$user->together('profile')->delete();
```