

27. 模型的多对多关联

学习要点：

1. 多对多关联

本节课我们来开始学习数据库模型多对多关联的方法。

二. 多对多关联

1. 多对多关联，比前面两种要复杂一些，需要一张中间表，共三张；

id	username	password	gender	id	type	id	user_id	role_id	details
19	蜡笔小新	123	男	1	超级管理员	1	19	2	(Null)
20	路飞	123	男	2	评论审核专员	2	20	4	(Null)
21	黑崎一护	456	男	3	图片监察员	3	21	5	(Null)
24	小明	123	男	4	帐户处理专员	4	24	1	(Null)
25	孙悟空	123	男	5	广告投放专员	5	25	3	(Null)

- (1) .users: 用户表;
- (2) .roles: 权限表;
- (3) .role_user: 中间表: 默认表名, user_id, role_id, 默认外键可不指明;

2. 创建权限表: Role.php, 留空; 在 User.php 设置多对多关联;

//多对多关联

```
public function role()
{
    return $this->belongsToMany(Role::class, 'role_user', 'user_id', 'role_id');
}
```

PS: 参数 2 传中间表名, 参数 3, 4 如果是默认值, 则可不传;

3. 多对多关联输出, 查看用户都拥有哪些权限;

```
$roles = User::find(19)->role;
return $roles;
```

4. 获取权限列表中某一个数据, 和一对多操作方法一样, 但注意返回的表名称;

```
//注意, 多对多这里 role()返回的是 role_user 表
//可以通过 dd($roles)查看, 所以, where 需要用 role_id 来指明
$roles = User::find(19)->role()->where('role_id', 1)->get();
return $roles;
```

//当然, 你也可以使用集合的方式去实现筛选

```
$roles = User::find(19)->role;
return $roles->where('id', 1);
```

5. 多对多的反向关联和其它两种方式也差不多；

//反向多对多关联，后面 id 是反的

```
public function user()
{
    return $this->belongsToMany(User::class, 'role_user', 'role_id',
    'user_id');
}

$users = Role::find(1)->user;
return $users;
```

6. 多对多会生成一个中间字段：pivot，里面包含多对多的双 id；
7. 如果想要 pivot 字段包含更多的中间表字段，可以自行添加，还可以修改字段名；

```
return $this->belongsToMany(Role::class, 'role_user', 'user_id', 'role_id')
    ->withPivot('details', 'id')
    ->as('pivot_name');
```

8. 定义多对多绑定时，可以在绑定方法内筛选数据；

```
$this->belongsToMany(Role::class) ... ->wherePivot('id', 1);
```

PS: 还有 wherePivotIn，以及派生的四种方法；

PS: 除了一对一，一对多，多对多，还有派生的远程一对一，远程一对多，以及多态一对一，多态一对多，多态多对多。这些更多的扩展，暂时不纳入基础的核心课程，防止过于繁杂冗余导致劝退。