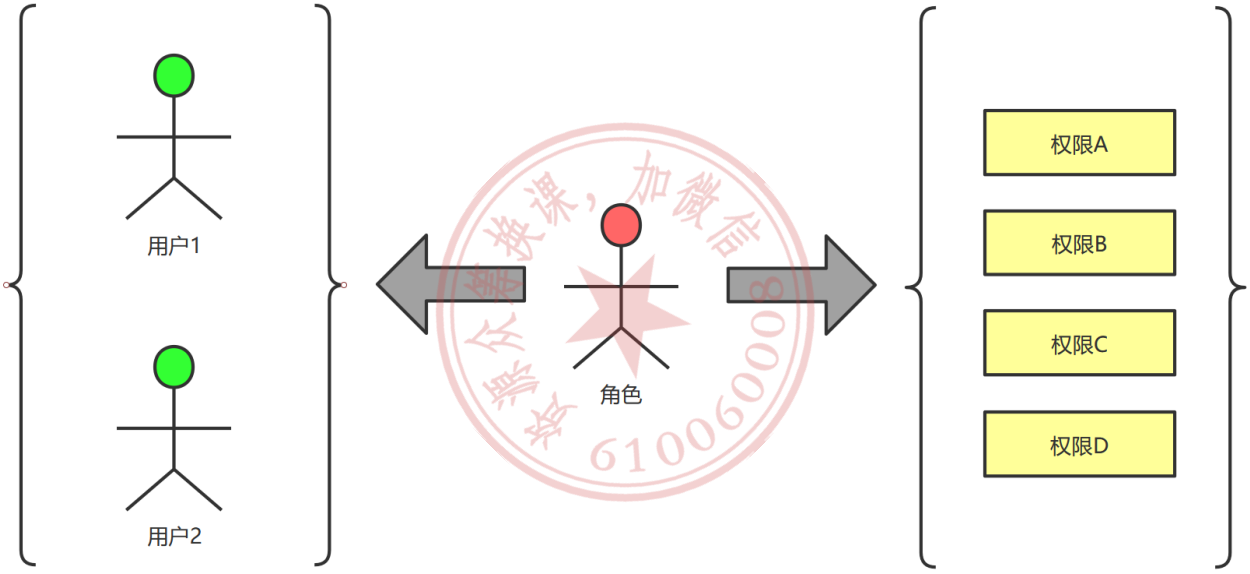
3-9 掌握RABC权限模型

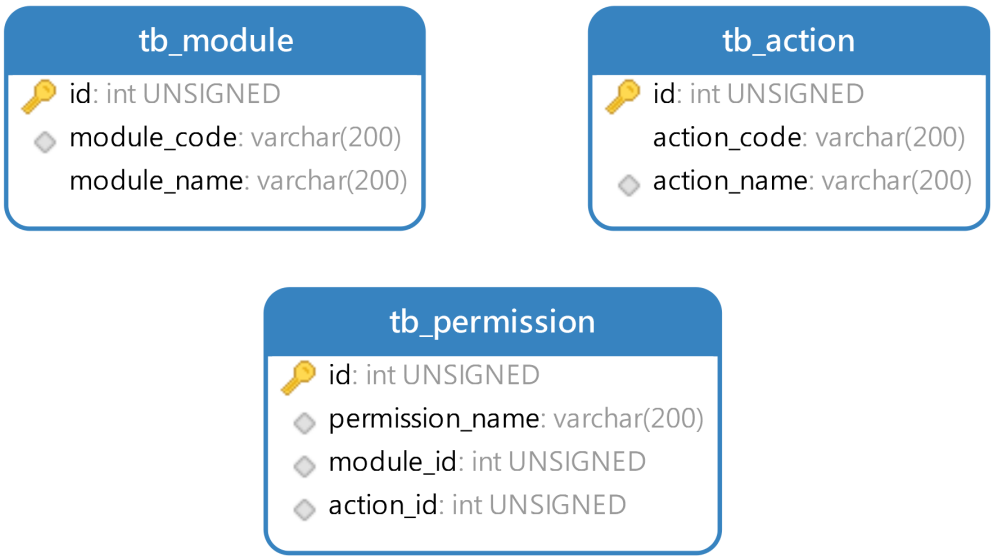
这个小节我们不先不急着写Web层的代码，因为注册成功之后，我们要向客户端返回令牌之外，还要返回用户的权限列表，以后客户端就可以根据权限列表判定用户能看到什么页面内容，以及可以执行什么操作，所以这个小节我们先来学习一下RBAC权限模型。

一、RBAC权限模型

RBAC的基本思想是，对系统操作的各种权限不是直接授予具体的用户，而是在用户集合与权限集合之间建立一个角色集合。每一种角色对应一组相应的权限。一旦用户被分配了适当的角色后，该用户就拥有此角色的所有操作权限。这样做的好处是，不必在每次创建用户时都进行分配权限的操作，只要分配用户相应的角色即可，而且角色的权限变更比用户的权限变更要少得多，这样将简化用户的权限管理，减少系统的开销。



RBAC模型中的权限是由模块和行为合并在一起而产生的，在MySQL中，有`模块表（tb_module）`和`行为表（tb_action）`，这两张表的记录合并在一起就行程了权限记录，保存在`权限表（tb_permission）`中。



现在知道了权限记录是怎么来的，下面我们看看怎么把权限关联到角色中。传统一点的做法是创建一个交叉表，记录角色拥有什么权限。但是现在 `MySQL5.7` 之后引入了 `JSON` 数据类型，所以我在 `角色表（tb_role）` 中设置的permissions字段，类型是JSON格式的。

id	role_name	permissions
0	超级管理员	[0]
1	总经理	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]
2	部门经理	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
3	普通员工	[1, 2, 3, 4, 5, 6, 7, 8]

到目前为止，JSON类型已经支持索引机制，所以我们不用担心存放在JSON字段中的数据检索速度慢了。MySQL为JSON类型配备了很多函数，我们可以很方便的读写JSON字段中的数据。

接下来我们看看角色是怎么关联到用户的，其实我在 `用户表（tb_user）` 上面设置role字段，类型依旧是JSON的。这样我就可以把多个角色关联到某个用户身上了。

id	open_id	nickname	sex	email	hiredate	role
3	oNlSI5C2Ih9(	神思者	(Null)	(Null)	(Null)	[0]

二、前后端权限验证

关于权限验证的工作，前端要做，后端也要做。后端的权限验证还好说，Shiro框架可以做这个事情。但是移动端没有权限验证框架，所以需要我们自己封装函数来验证权限。每个页面在渲染的时候，先判断用户拥有什么权限，然后根据权限控制渲染的内容。比如说普通员工没有添加新员工的权限，所以界面上就不能出现添加按钮。

移动端做权限判断的前提是必须有当前用户的 `权限列表` ，这个权限列表是用户 `登陆成功` 或者 `注册成功` ，后端Java项目返回给移动端的，移动端保存到本地 `Storage` 里面。

三、如何查询用户的权限列表？

```
SELECT p.permission_name
FROM tb_user u
JOIN tb_role r ON JSON_CONTAINS(u.role, CAST(r.id AS CHAR))
JOIN tb_permission p ON JSON_CONTAINS(r.permissions, CAST(p.id AS CHAR))
WHERE u.id = 用户ID AND u.status = 1;
```

在`TbUserDao.xml`文件中添加上面的SQL语句，用来查询用户的权限列表

```
<select id="searchUserPermissions" parameterType="int" resultType="String">
    SELECT p.permission_name
    FROM tb_user u
    JOIN tb_role r ON JSON_CONTAINS(u.role, CAST(r.id AS CHAR))
    JOIN tb_permission p ON JSON_CONTAINS(r.permissions, CAST(p.id AS CHAR))
    WHERE u.id = #{userId} AND u.status = 1;
</select>
```

在`TbUserDao.java`接口中声明`searchUserPermissions()`方法

```
public Set<String> searchUserPermissions(int userId);
```

在`UserService.java`接口中声明`searchUserPermissions()`方法

```
public Set<String> searchUserPermissions(int userId);
```

在`UserServiceImpl.java`接口中实现`searchUserPermissions()`方法

```
@Override
public Set<String> searchUserPermissions(int userId) {
    Set<String> permissions=userDao.searchUserPermissions(userId);
    return permissions;
}
```