

**实验报告**

课 程 名 称 ： 数据库系统原理

实 验 题 目： 创建和删除数据库

学生所在专业： 计算机科学与技术

学生所在班级： 计算机1802班

学 生 学 号 ： 1816050215

学 生 姓 名 ： 杨淋钧

实 验 日 期 ： 2020.5.25

指导教师姓名： 张洋

实 验 成 绩 ：

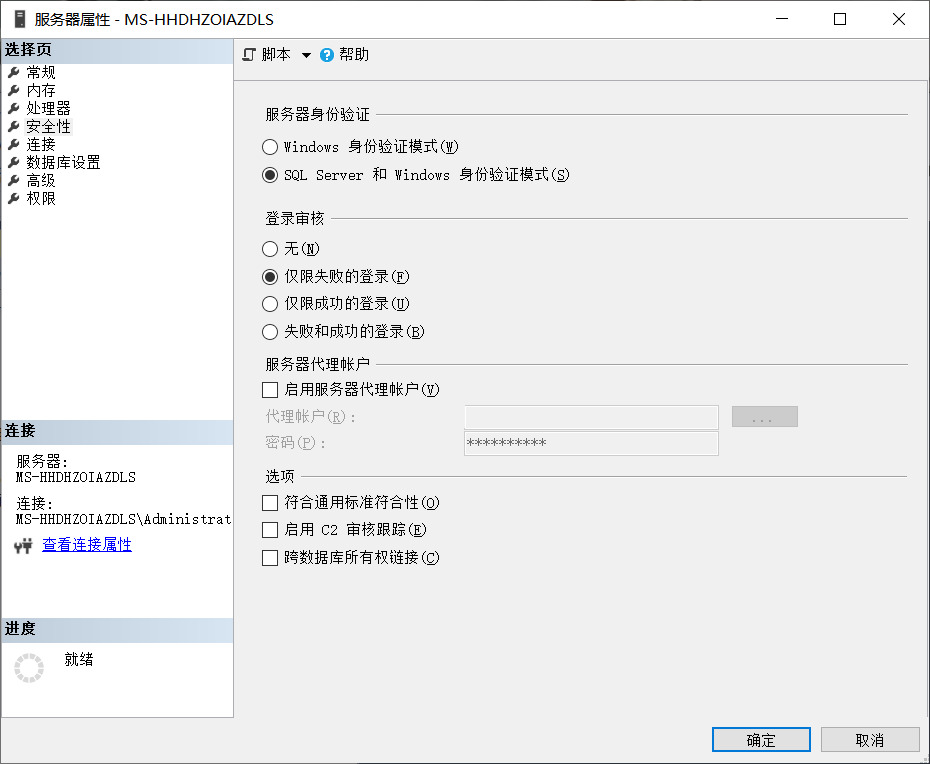
# 实验四 数据库安全性实验

**一、实验目的**

1. 理解安全性的概念。
2. 在理解用户及相关概念的基础上掌握自主存取控制机制（DAC）。
3. 熟悉MS SQL SERVER的安全性技术。

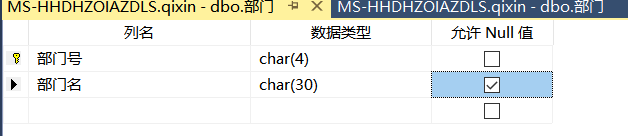
**二、实验内容及步骤**

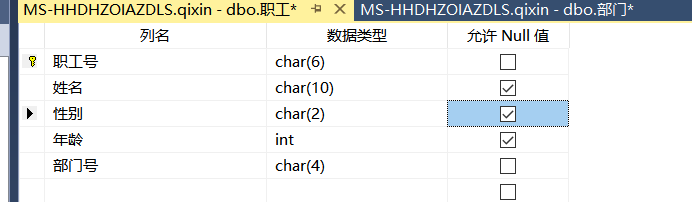
**1**设置MS SQL Server认证模式，可以用在Microsoft SQL Server Management Studio中选择服务器节点，用右键单击服务器的弹出菜单选择“属性”，如图1，点击“安全性”页。



1. 创建数据库qixin

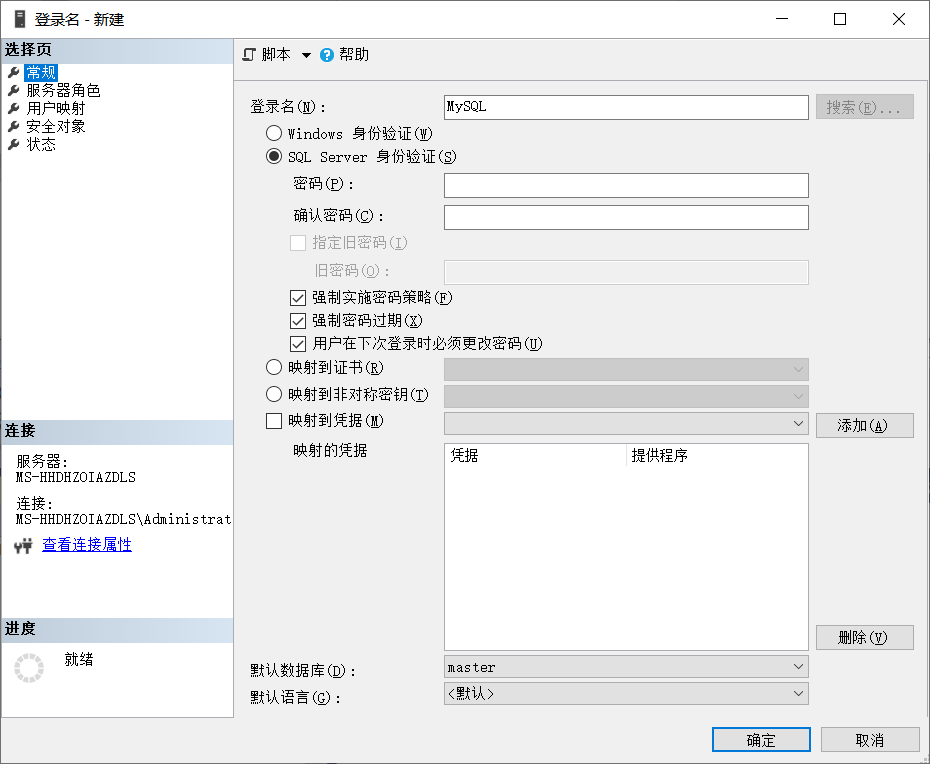


包括两张表，职工表和部门表

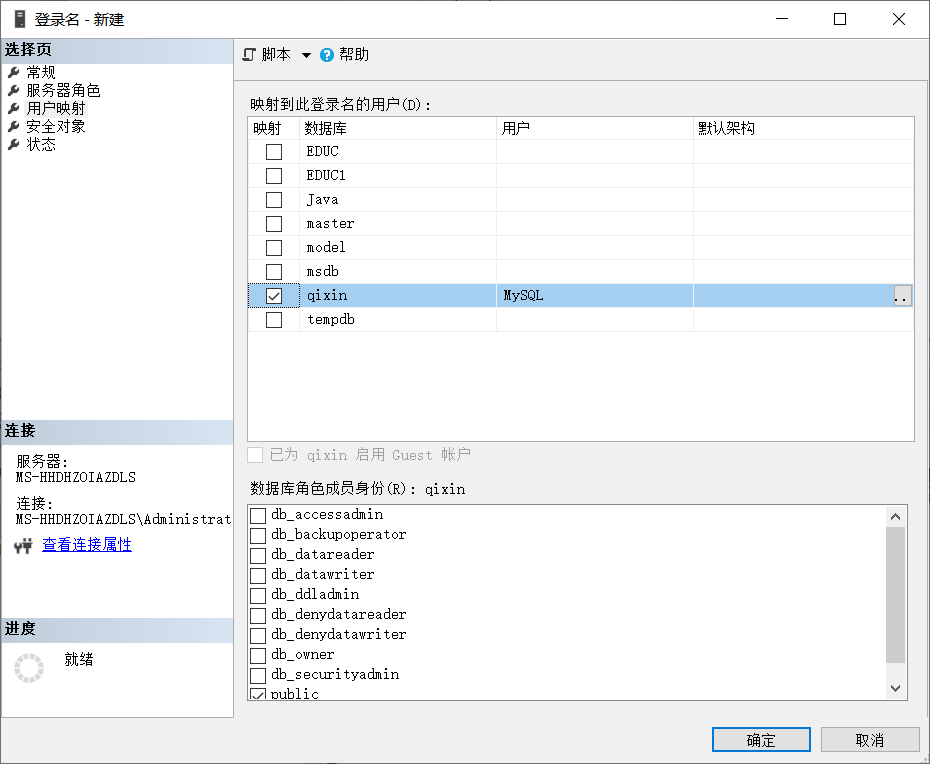


1. 定义登陆与用户

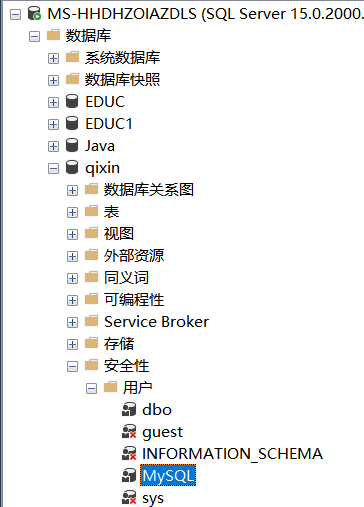
方法一：图形界面：通过Microsoft SQL Server Management Studio。在选择服务器的“安全性”－> “登录名”右键“新建登录名”，输入登录名称logqixin，选择SQL Server身份验证，如图2。

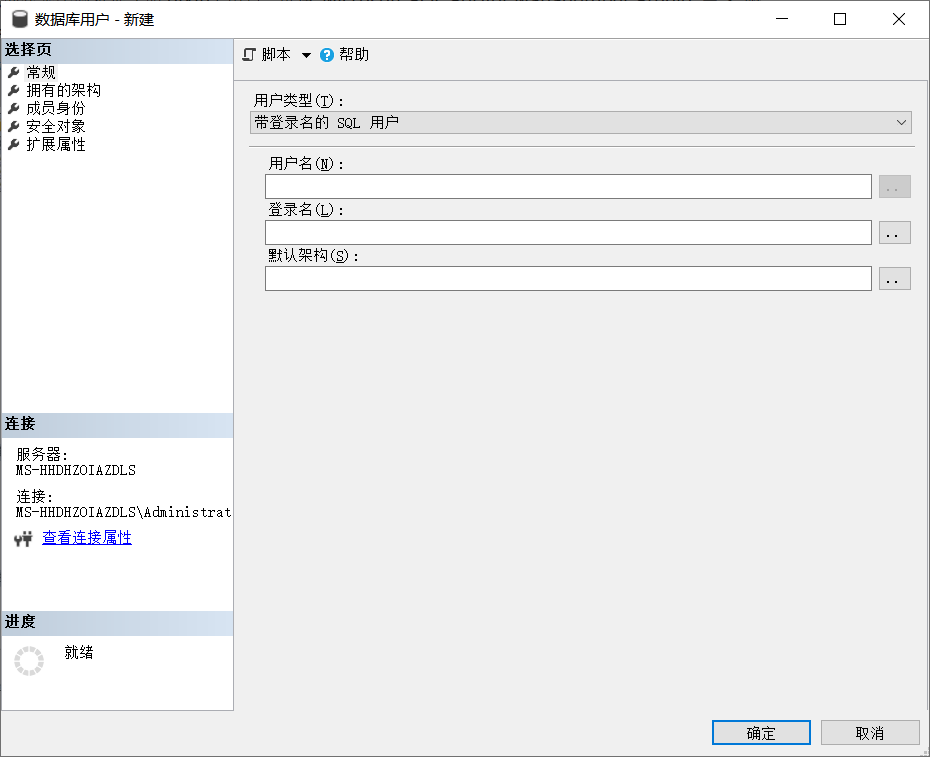


单击“用户映射”页面，注意允许访问数据库“qixin”，如图3。



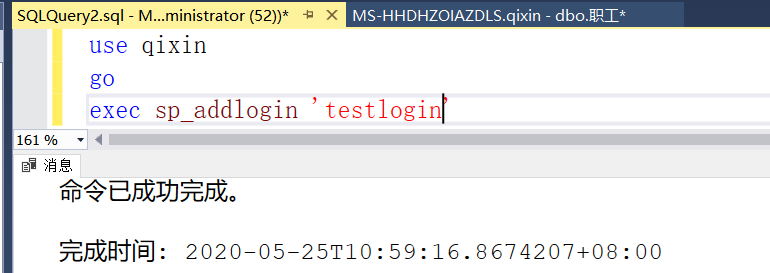
此时，SQLSERVER会自动在数据库qixin的“用户”中添加一个与登陆同名的用户logqixin。这样登陆logqixin与同名用户logqixin（访问数据库qixin）就建立起来了。 如图4



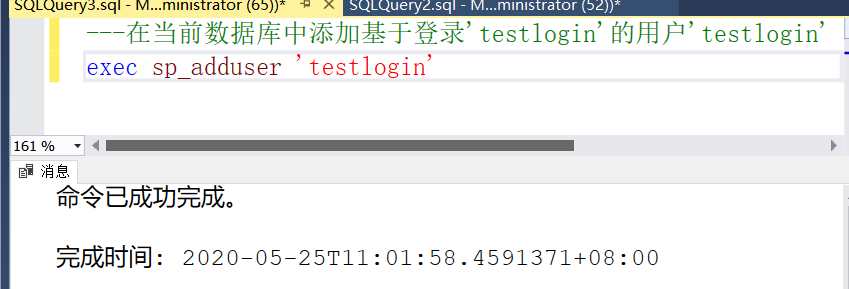
如果单独创建某个数据库（如qixin）用户，可在“Microsoft SQL Server Management Studio”－> 服务器 －> “数据库” －> “用户”右键单击用户的弹出菜单选择“新建数据库用户”，如图5。(注意，在此之前必须先建立一个登录，然后新建用户与此登录关联。)

方法二：通过系统存储过程

1. 添加登陆，如图6。



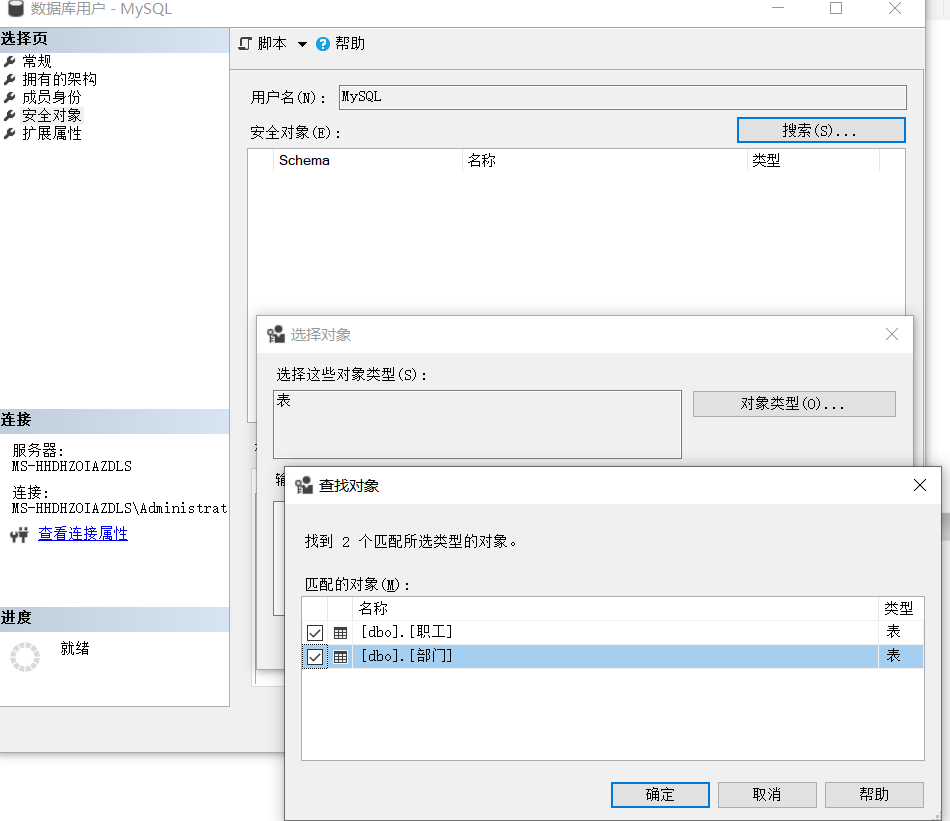
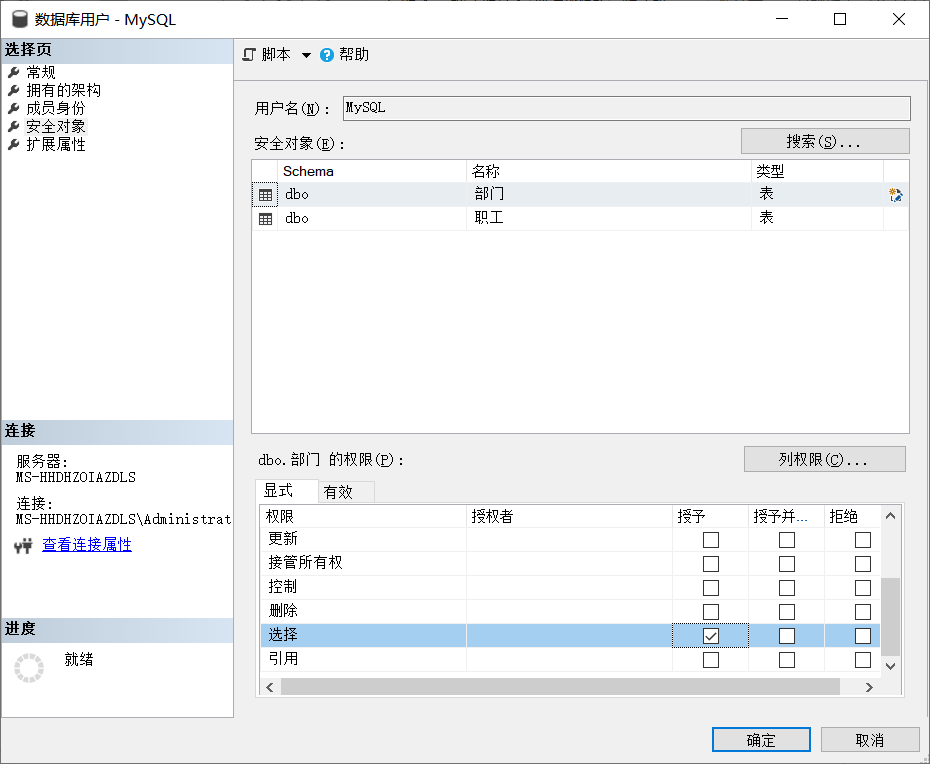
（2）添加用户，如图7。



4、完成授权/撤销权限。观察授权前后的变化（以表“部门”与用户“logqixin”为例）。

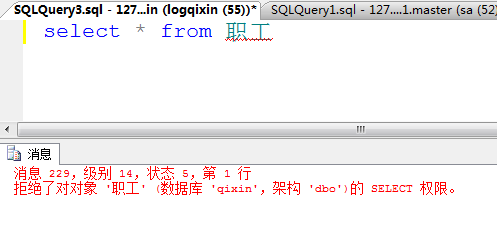
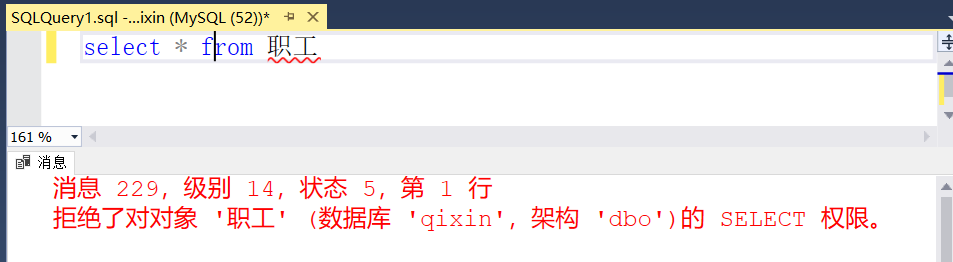
（1）授权

方法一：图形界面，通过“Microsoft SQL Server Management Studio” －> 服务器 －> “数据库” －> “用户” －> “logqixin”－>右键“所有任务” －> “管理权限”，如图8。

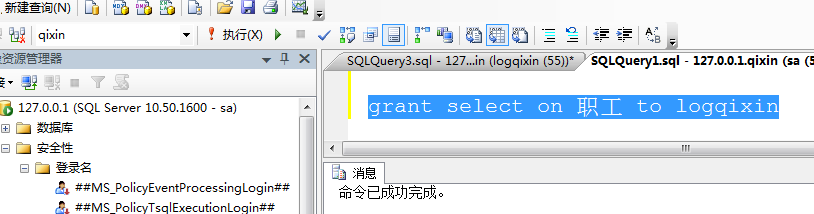


方法二：GRANT命令。

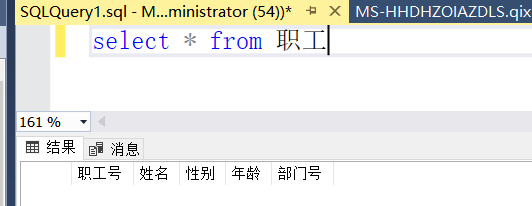
以“logqixin” 连接数据库“qixin”，执行查询，如图9。



1. 以“sa”连接数据库qixin，并执行授权，如图10。



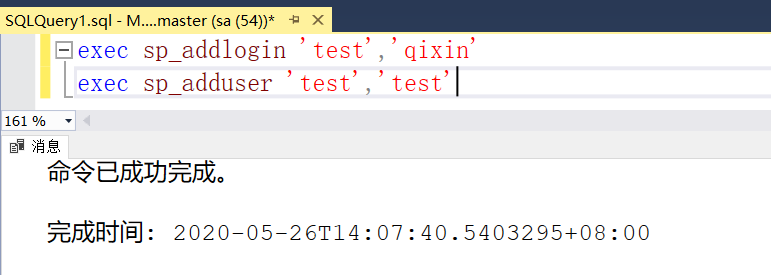
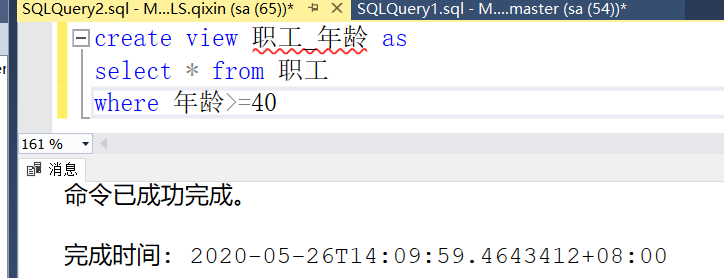
（5）再以”logqixin”连接qixin数据库，执行查询，结果显然不同，见图11。



5、设计安全机制使得用户“test”只能查询年龄在40岁以上（包括）职工。

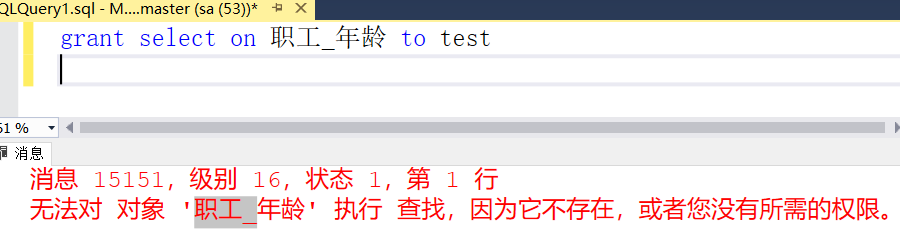
（1）如果用户“test”不存在的话，按照前述方法定义用户“test”。

exec sp\_addlogin 'test','','qixin'  
exec sp\_adduser 'test','test'



（2）创建基于表“职工”年龄在40岁以上（包括）的职工视图。

create view 职工\_年龄 as   
select \* from 职工  
where 年龄>=40

1. 将职工\_年龄视图查询权限授予用户“test”。

grant select on 职工\_年龄 to test

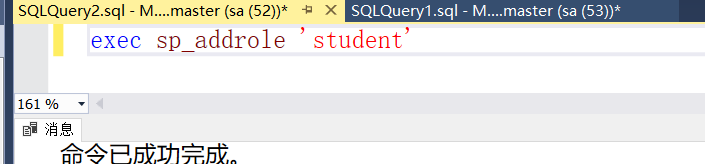
6、设计角色“student”，可以查看“职工”的职工号与姓名。

角色（ROLE）：一组权限的集合。数据库管理员将数据库的权限赋予角色，然后将角色再赋予给数据库用户或登录帐户，从而使数据库用户或登录帐户拥有相应的权限。

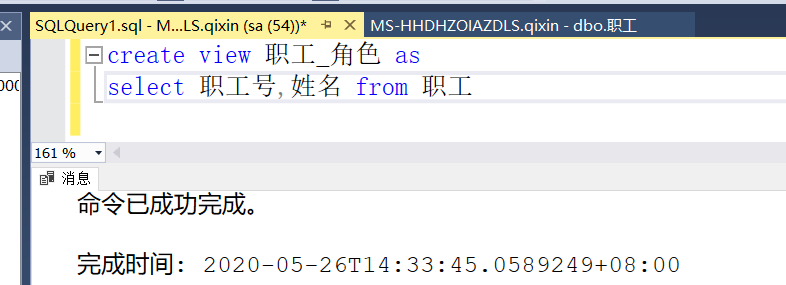
MS SQL Server提供了固定服务器角色和数据库角色，用户可以修改固定数据库角色的权限，也可以自己创建新的数据库角色，再分配权限给新的角色。除了系统预定义的角色外也可以定义（CREATE ROLE）。public 角色是一个特殊的数据库角色，每个数据库用户都属于它。 role

（1）定义角色“学生”。

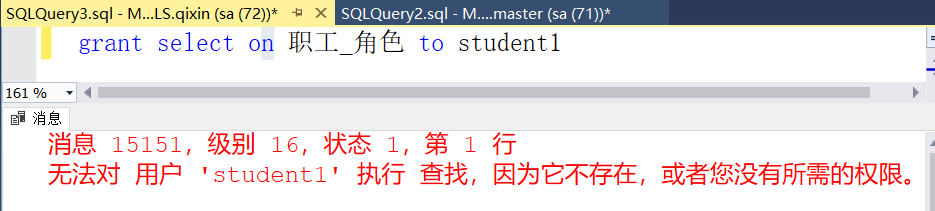
exec sp\_addrole 'student'



（2）创建基于表“职工”且只包含“职工”的职工号与姓名的职工视图。

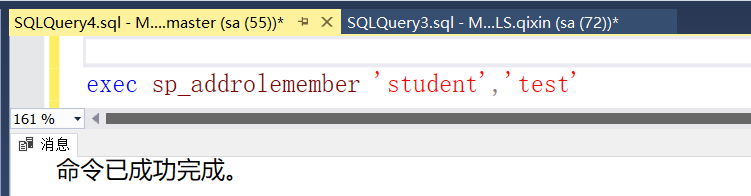
create view 职工\_角色 as   
select 职工号,姓名 from 职工

（3）将只包含“职工”的职工号与姓名的职工视图查询权限授予角色“student”。

grant select on 职工\_角色 to student

（4）将用户“test”作为成员加入角色“学生”，这样用户“test”只能查看“职工”的职工号与姓名。

--添加用户test到角色student中  
exec sp\_addlogin 'test'  
exec sp\_adduser 'test','test'  
exec sp\_addrolemember 'student','test'



**三．实验总结**

1 ．实验中遇到的问题及解决过程

跟着老师的步骤去进行试验，感觉是比较简单的事情，也会遇到比如说，存在错误的命令，不符合SQL语句的标准，用了一些中文的命令导致出现的报错，还有就是在属性设置SQL身份验证是出现的错误等等。

2 ．实验中产生的错误及原因分析

使用标准的SQL语句以及学会不同的操作用什么样的命令，千万不要使用中文的符号，然后还有一个身份报错的原因是没有给与启动SQL身份验证的权限而导致的错误。

3 ．实验体会和收获。