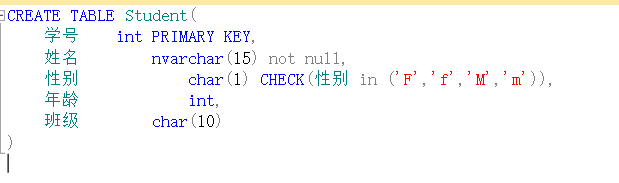
1.1. 实验三 SQL Server 安全机制

实验目的

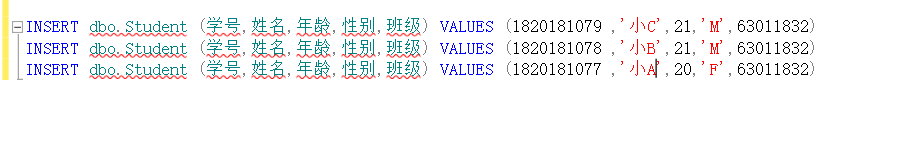
通过本实验，掌握SQL Server安全机制，包括SQL Server安全、数据库用户与权限，掌握身份验证、登录、数据库用户、服务器角色、数据库角色、权限与授权方法等相关概念及其相互关系。

实验内容

（1） 创建一个数据库。在新创建的数据库下创建一个学生基本信息表，插入若干记录。



首先，先创建一个数据库表，在里面添加了学号，姓名，性别，年龄，班级这五个项目。

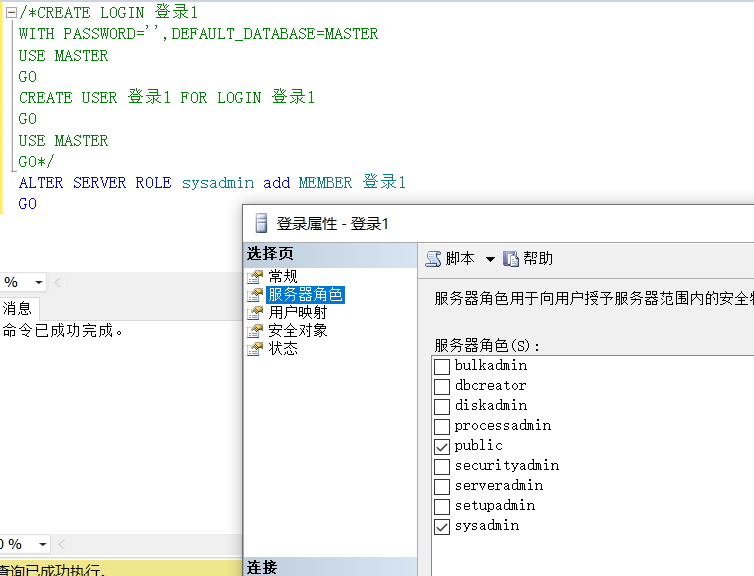


姓名性别和班级使用char类型输入，因此在输入时需要用到单引号。



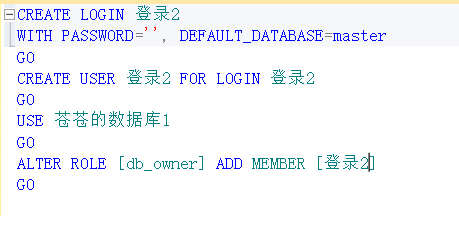
已插入三条数据。

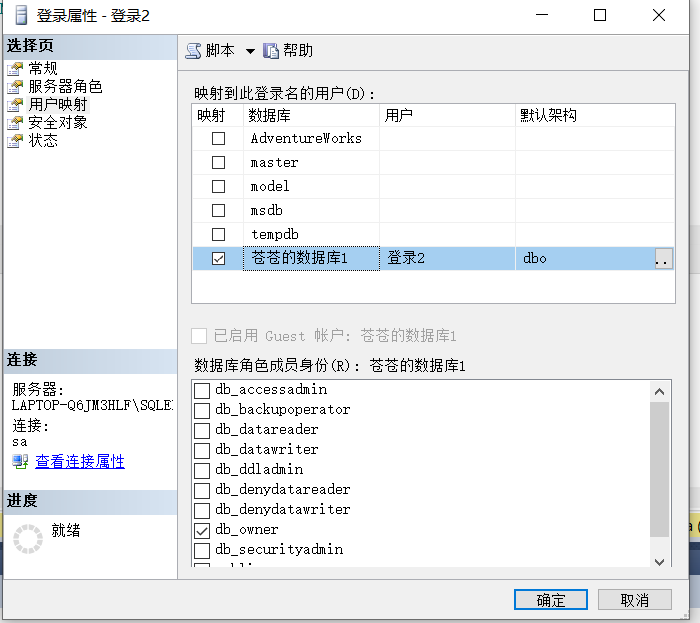
（2） 创建一个Windows身份验证的登录1，将默认数据库设置为Master数据库。将登录1设置为数据库系统管理员。



在账户SA下面创建一个Windows身份验证登录1，将登录1默认存在于MASTER数据库下面，并使其服务器角色成为sysadmin。

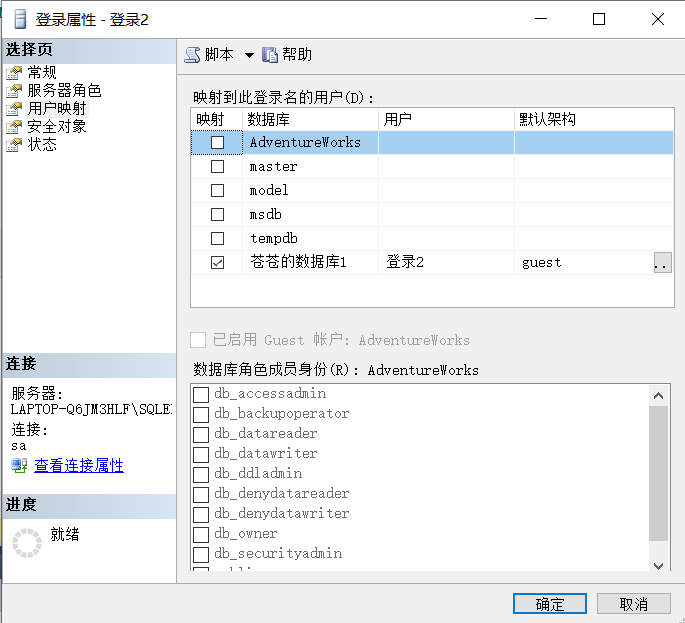
（3） 创建一个Windows身份验证的登录2，将默认数据库设置为新创建的数据库。授权登录2为新创建的数据库的管理员。





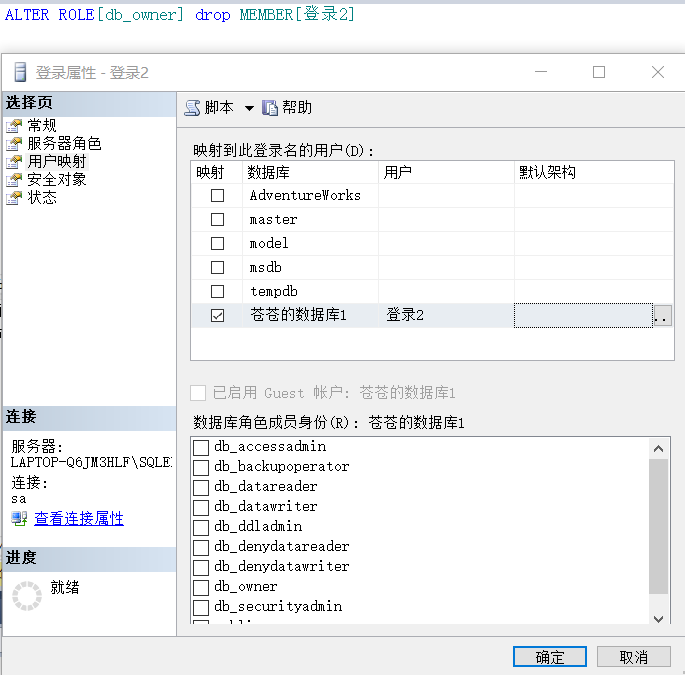
这一步的代码和上一段代码差不多，唯一需要更改的地方就是新创一个数据库（苍苍的数据库1），并将登录2默认存在于该数据库下，授权登录2为新创建的数据库（苍苍的数据库1）的管理员，最后让服务器角色sysadmin改为数据库角色db\_owner。

（4） 撤消登录2的新创建的数据库的管理员权限。



这一块我实验了两个方法：

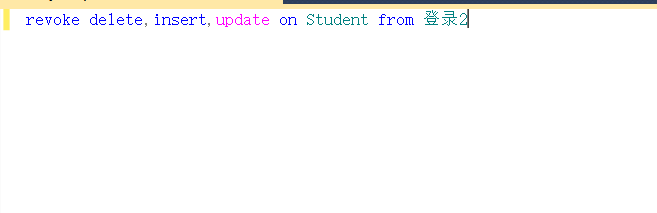
第一个是将用户登录2的默认架构设置为guest，使其撤销管理员权限。



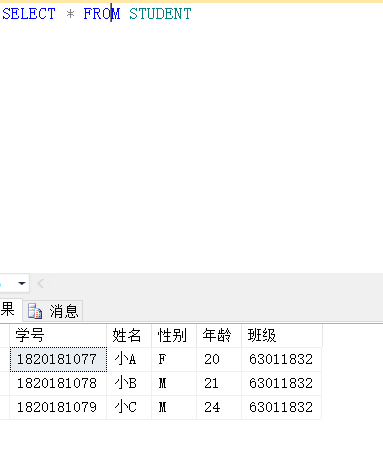
第二个方法是利用ALTER ROLE[db\_owner] drop MEMBER[登录2]

这段代码来使登录2撤销默认架构，使其失去管理员的权限。

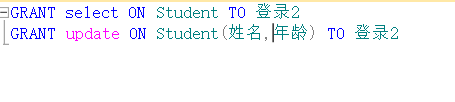
（5） 授权登录2查询和更新学生基本信息表中部分字段的权限。验证查询、有权限更新、无权限更新、无权限插入和删除。



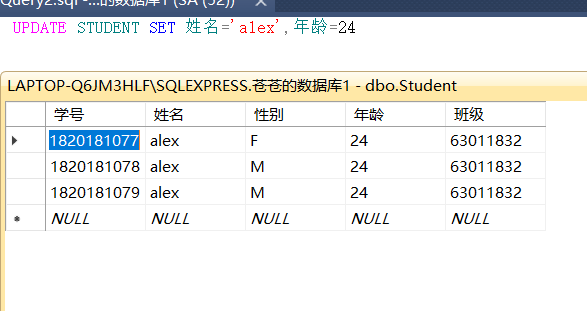
第一步先将登录2的删除添加更新权限全部回收



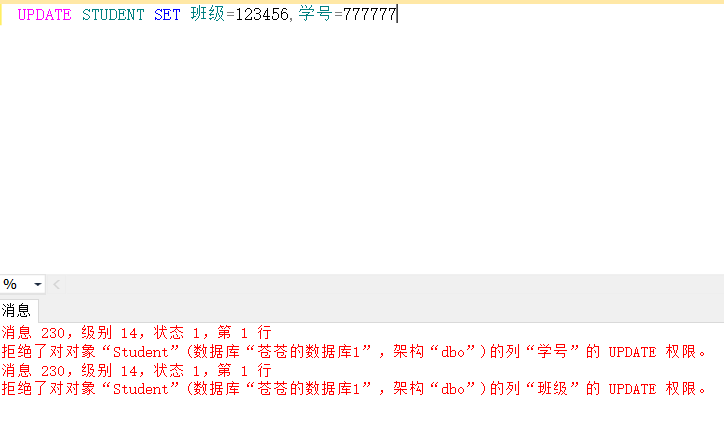
验证查询表中的数据



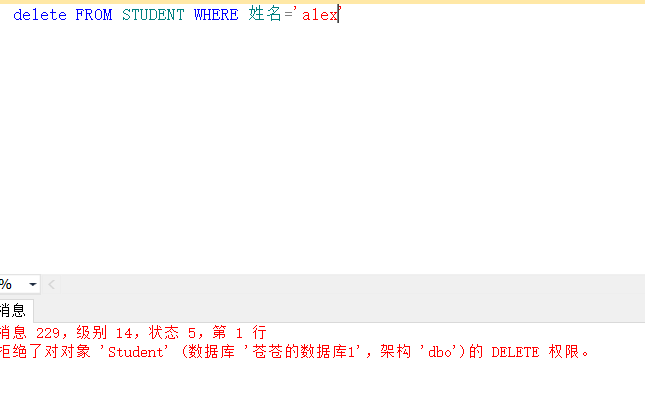
授予部分更新权限给登录2，这里授予可更新的权限为姓名和年龄。



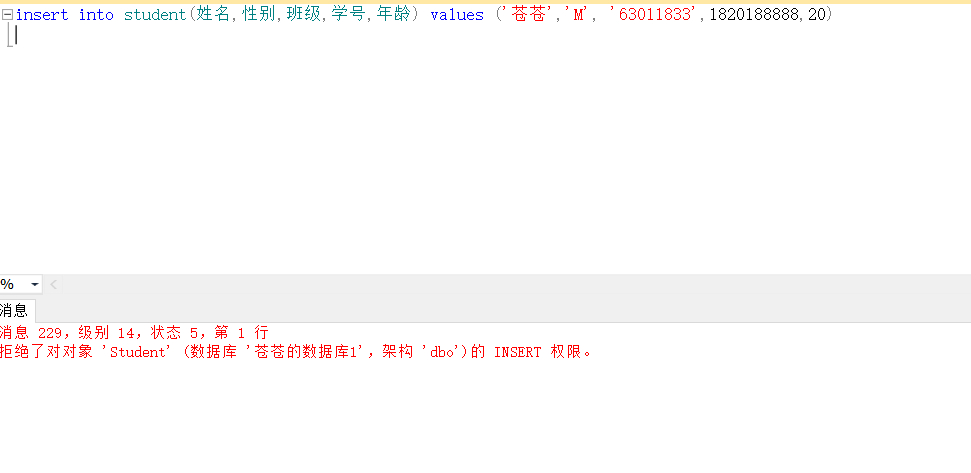
如上图可见，修改后的表格姓名全部变成alex，而年龄则全部变为24岁。



由于没有赋予学号和班级的权限，因此在修改该部分数据时被拒绝对象。



由于回收了删除的权限，所以当我们试图删除一条数据时，消息提示拒绝对象的DELETE权限。

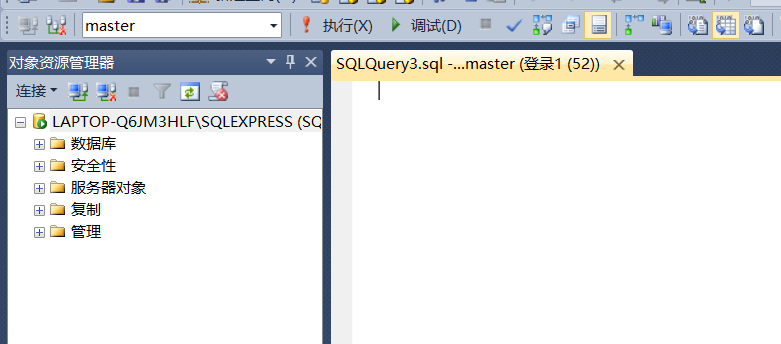


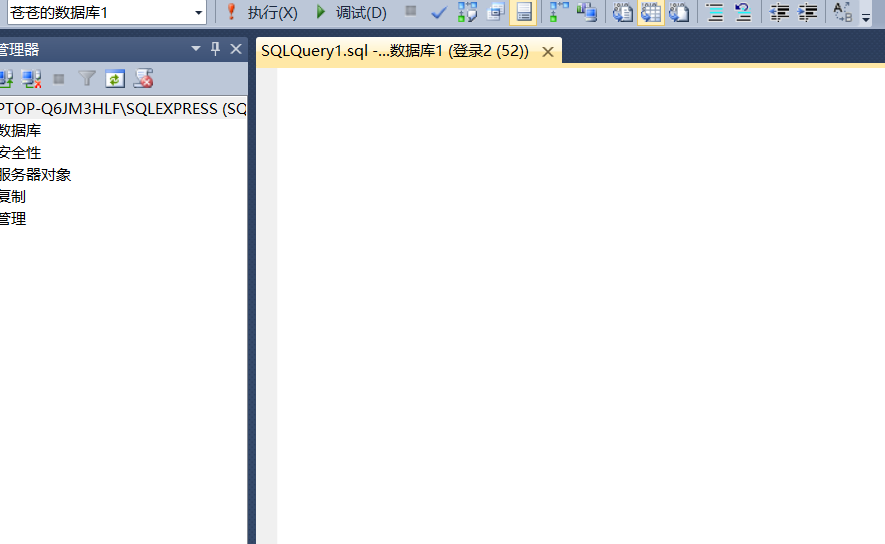
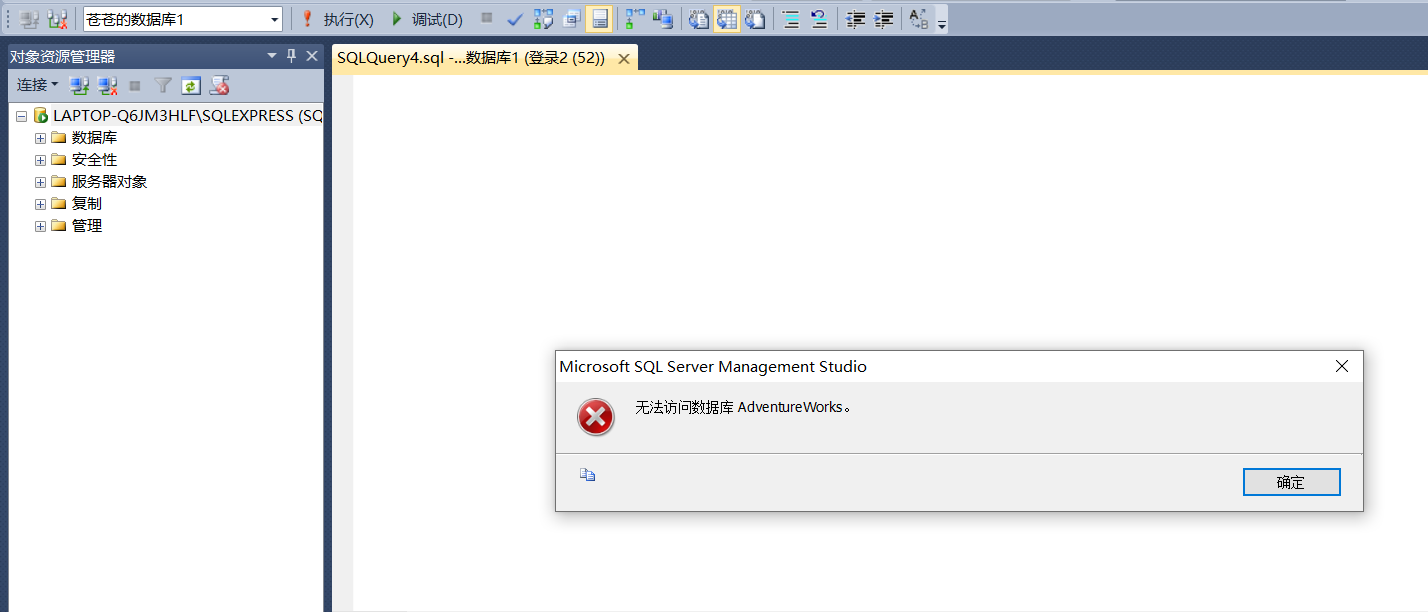
由于回收了添加的权限，所以当我们试图添加一条新的数据进入数据库时，消息提示拒绝对象的INSERT权限。

回答下列问题

1. 简述什么是默认数据库并用实验结果说明不同的默认数据库在登录到数据库服务器后有什么不同。

当用户登录时默认连接到的数据库就是默认数据库。比如用登陆1账户连接进入时的默认状态显示的是MASTER，而用登录2账户连接进入时的默认状态显示是自己建立的数据库，当想要将登录2切换到MASTER时系统显示无权限，这是因为在创建登录2账户时没有将MASTER添加进登录2账户所导致的，因此当想要试图让登录2获得MASTER数据库时，系统会提示无法访问。



（2） 说明实验内容（2）、（3）、（5）中所用的授权方法的差异，简述每种方法的优缺点。

实验内容（2）

优点：

这个授权方法是将登录1授权为数据库系统管理员sysadmin，sysadmin具有着对所有的数据库的全部权限，可以任意修改和操作数据库。

缺点：

由于这个方法的授权十分简单，如果授权数量太多的话，会对数据库产生非常大的危险，比如任意修改操作权限的话可能会导致某些不能修改的权限被修改，从而导致数据库无法正常运行。

实验内容（3）

优点：

这个授权方法是将登录2授权为新创建的数据库的管理员，并让该数据库角色变成db\_owner，登录2仅对这个新创建的数据库有全部权限，可以任意操作这个数据库，这种方法对不同用户可以授予不同的数据库权限。

缺点：

和实验内容（2）的缺点一样，这个授权方法授权的登录2拥有在新创建的数据库内的所有修改权限，对于该数据库来说十分不安全。

实验内容（5）

优点：

这个授权方法是将登陆2授于其部分权限，使登录2并非具有完全修改数据库所有权限的能力，这样的授权方法对于数据库的安全性会比上面两种要安全得多。

缺点：

授权方法十分复杂，需要用的grant和revoke来进行部分授权和回收，且对于不同的权限还有不同的操作方法，十分繁琐。