

****

**J I A N G S U U N I V E R S I T Y**

**“数据库系统原理”实验报告**

|  |  |
| --- | --- |
| 学院名称： | 计算机科学与通信工程学院 |
| 专业班级： | 物联网1801 |
| 学生姓名： | 张 承 楷 |
| 学生学号： | 3180611023 |
| 指导教师： | 邢 玉 萍 |

**2020年11月 26日**

目录

[实验一 数据库的定义与建立实验 1](#_Toc57555549)

[一、实验目的 1](#_Toc57555550)

[二、实验内容 1](#_Toc57555551)

[三、实验实现及结果 1](#_Toc57555552)

[四、注意事项 5](#_Toc57555553)

[五、思考题 5](#_Toc57555554)

[实验二 数据库的查询实验 6](#_Toc57555555)

[一、实验目的 6](#_Toc57555556)

[二、实验内容 6](#_Toc57555557)

[三、实验实现及结果 6](#_Toc57555558)

[四、注意事项 9](#_Toc57555559)

[五、思考题 9](#_Toc57555560)

[实验三 数据库的维护实验 10](#_Toc57555561)

[一、实验目的 10](#_Toc57555562)

[二、实验内容 10](#_Toc57555563)

[三、实验实现及结果 10](#_Toc57555564)

[四、注意事项 12](#_Toc57555565)

[五、思考题 12](#_Toc57555566)

[六、补充题 13](#_Toc57555567)

[实验四 数据库的安全性、完整性实验 14](#_Toc57555568)

[一、实验目的 14](#_Toc57555569)

[二、实验内容 15](#_Toc57555570)

[三、实验实现及结果 15](#_Toc57555571)

**实验一 数据库的定义与建立实验**

**一、实验目的**

要求学生熟练掌握和使用T－SQL﹑SQL Server企业管理器创建数据库﹑基本表﹑视图﹑索引和修改表结构，及向数据库输入数据的操作；学会创建和使用表的主外码和约束。

**二、实验内容**

针对上述题目1零件供应数据库应用系统分别完成以下相应任务：

1. 创建相应的数据库和查看数据库属性。

2. 创建基本表﹑确定表的主码和相应的约束，为主码建索引。

3. 创建﹑查看视图。

4. 创建表之间的关联。

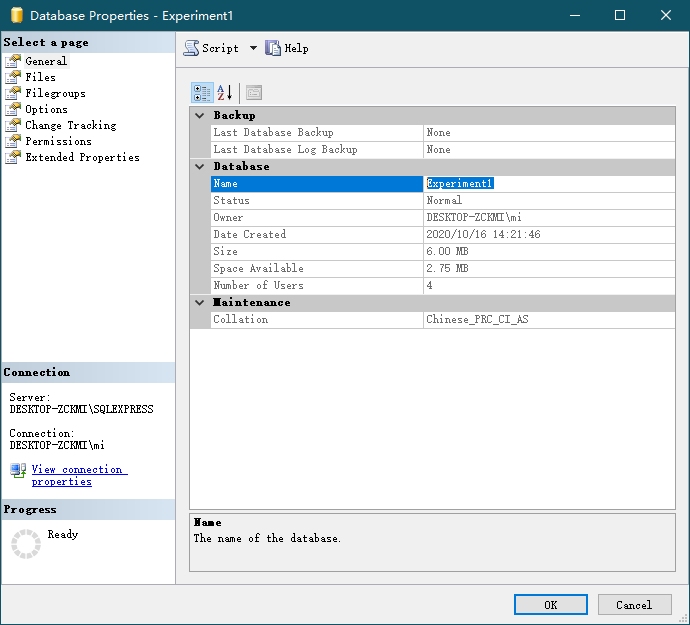
5. 利用T－SQL和SQL Server企业管理器向数据库输入数据。

**三、实验实现及结果**

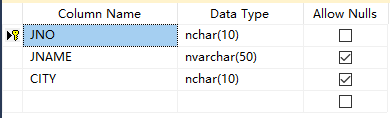
某供应商（S）供应（SPJ）某种零件（P）给某个工程项目（J），其中一个供应商可以供给多个项目零件，而每个项目可以使用多个供应商供应的零件，每种零件可由不同供应商供给。

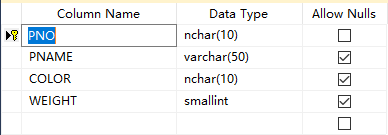
本实验针对上述题目分别完成以下相应任务：

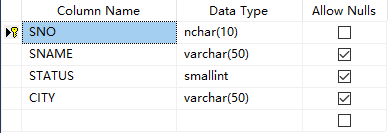
1﹑创建相应的数据库和查看数据库属性；

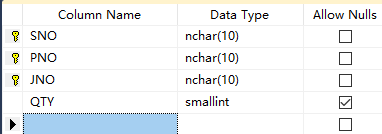


2﹑创建基本表﹑确定表的主码和相应的约束，为主码建索引；

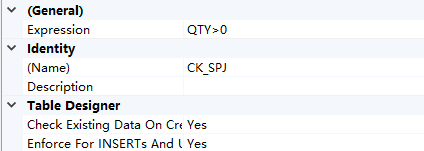


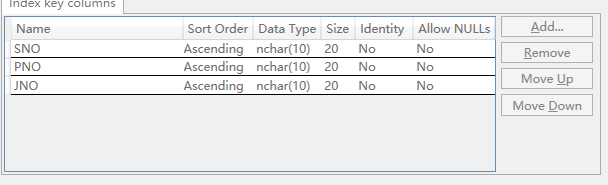




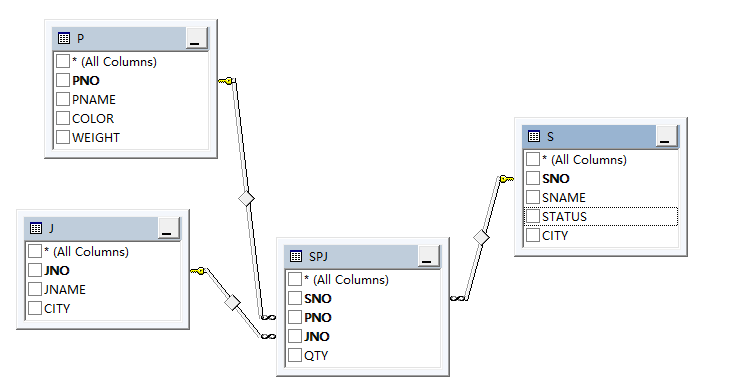








3﹑创建﹑查看视图；



SELECT dbo.J.JNO, dbo.J.JNAME, dbo.J.CITY, dbo.P.PNO, dbo.P.PNAME, dbo.P.WEIGHT, dbo.P.COLOR, dbo.S.SNAME, dbo.S.SNO, dbo.S.STATUS, dbo.S.CITY AS Expr1, dbo.SPJ.SNO AS Expr2, dbo.SPJ.PNO AS Expr3, dbo.SPJ.QTY, dbo.SPJ.JNO AS Expr4

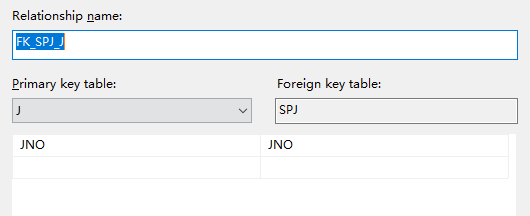
FROM dbo.J INNER JOIN

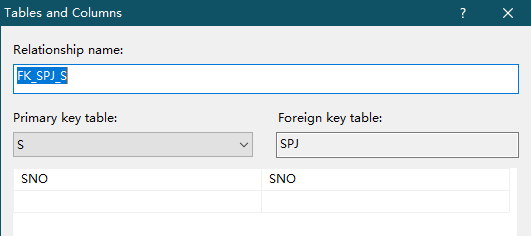
dbo.SPJ ON dbo.J.JNO = dbo.SPJ.JNO INNER JOIN

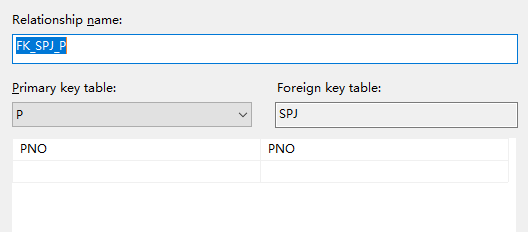
dbo.P ON dbo.SPJ.PNO = dbo.P.PNO INNER JOIN

dbo.S ON dbo.SPJ.SNO = dbo.S.SNO

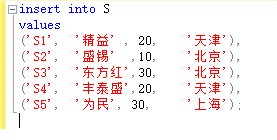
4﹑创建表之间的关联；

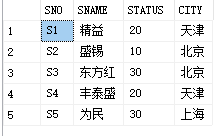






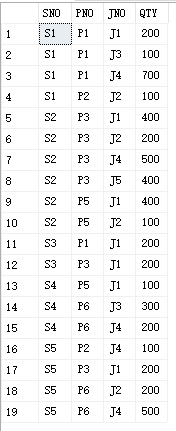
5﹑利用T－SQL和SQL Server企业管理器向数据库输入数据。











**四、注意事项**

1. 输入数据时要注意数据类型、主码和数据约束的限制。

2. 注意数据库的主码﹑外码和数据约束的定义。

3. 参照表和被参照表之间的关系，主码和外码间的关系。

**五、思考题**

1﹑为什么要建立索引？在你的数据库中建立多少索引合适？

答：（1）在关系数据库中，索引是一种单独的、物理的对数据库表中一列或多列的值进行排序的一种存储结构，它是某个表中一列或若干列值的集合和相应的指向表中物理标识这些值的数据页的逻辑指针清单，索引帮助数据库高效获取数据，有了索引数据库可以通过二分算法快速查找。

（2）根据数据库管理员根据实际的用途以及数据库中记录的情况，来进行判断，需要在更新速度与查询速度之间取得一个均衡点。

2﹑索引和视图能否修改吗？为什么？

答：（1）索引可以修改，通过ALTER INDEX语句来停用索引，重建索引，整理索引或修改索引选项。

（2）视图可以修改，通过ALTER VIEW语句来修改，或者使用视图来修改

3﹑为什么不能随意删除被参考表中的主码？

答：为了保证实体完整性与参照完整性。

**实验二 数据库的查询实验**

**一、实验目的**

1. 要求学生掌握SQL Server查询分析器和T－SQL语言的使用方法，熟练掌握和使用简单表的数据查询﹑数据排序和数据连接查询﹑嵌套查询等操作方法。

2. 熟练掌握和使用分组查询和函数查询的使用方法，完成数据查询中的统计、计算输出的操作方法。

**二、实验内容**

1. 检索使用上海产的零件的工程名称。

2. 检索供应工程J1零件P1的供应商号SNO。

3. 检索供应工程J1零件为红色的供应商号SNO。

4. 检索没有使用天津生产的红色零件的工程号JNO。

5. 检索至少用了供应商S1所供应的全部零件的工程号JNO。

6. 检索购买了零件P1的工程项目号JNO及数量QTY，并要求对查询的结果按数量QTY降序排列。

7. 找出向北京供应商购买重量大于30的零件的工程号。

8. 找出工程项目J2使用的各种零件的名称及其数量。

9. 按工程号递增的顺序列出每个工程购买的零件总数。

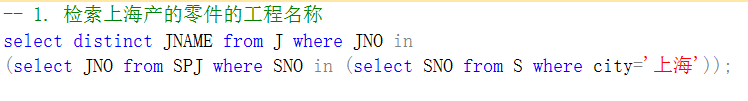
10. 编程输出如下报表：

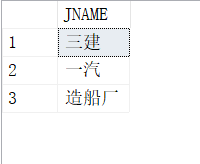
表2.1 报表

|  |  |  |  |
| --- | --- | --- | --- |
| 供应商 | 零件 | 工程项目 | 数量 |

**三、实验实现及结果**

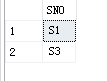
1. 检索上海产的零件的工程名称；





2. 检索供应工程J1零件P1的供应商号SNO；



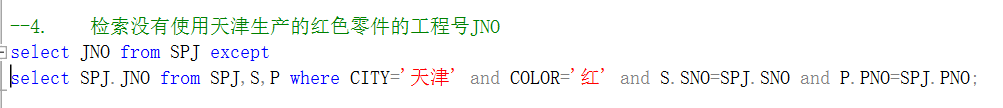


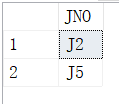
3. 检索供应工程J1零件为红色的供应商号SNO；



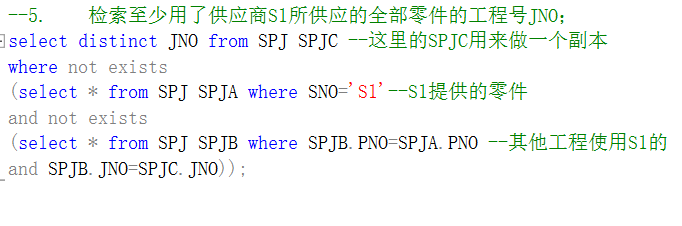


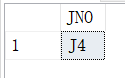
4. 检索没有使用天津生产的红色零件的工程号JNO；



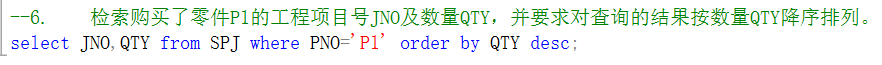


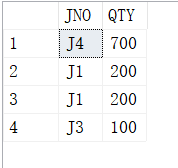
5. 检索至少用了供应商S1所供应的全部零件的工程号JNO；



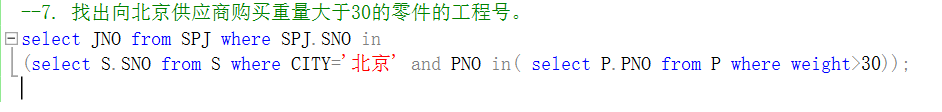


6. 检索购买了零件P1的工程项目号JNO及数量QTY，并要求对查询的结果按数量QTY降序排列。



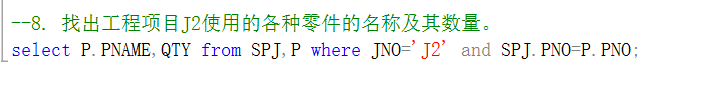


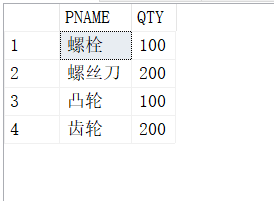
7. 找出向北京供应商购买重量大于30的零件的工程号。



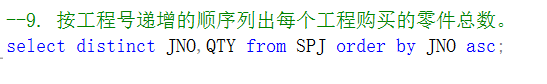


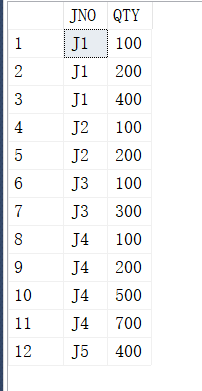
8. 找出工程项目J2使用的各种零件的名称及其数量。





9. 按工程号递增的顺序列出每个工程购买的零件总数。

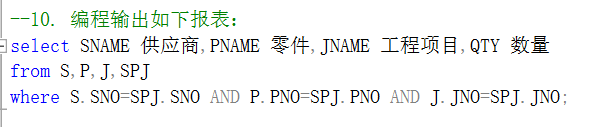


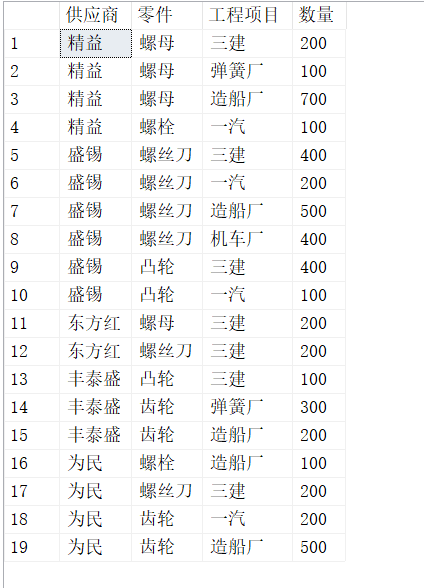


10. 编程输出如下报表：

表2.1 报表

|  |  |  |  |
| --- | --- | --- | --- |
| 供应商 | 零件 | 工程项目 | 数量 |





**四、注意事项**

1. 内连接、左外连接和右外连接的含义及主要方法。

2. 输入SQL语句时应注意，语句中均使用西文操作符号。

3. 子句WHERE（条件）表示元组筛选条件，子句HAVING（条件）表示分组选择条件。

4. 子句HAVING（条件）必须和GROUP BY（分组字段）子句配合使用。

**五、思考题**

1. 如何提高数据查询和连接速度？

答：（1）硬件升级，优化网络（2）恰当地使用索引，使用缓存（3）优化SQL语句，如减少连接查询和排序。

2. 试比较连接查询和嵌套查询。

答：（1）嵌套查询是包含一个或多个子查询或子查询的另一个术语的select语句。在外部查询中包含内部查询的另一个子查询是SQL语句的扩展，嵌套查询可以用连接来代替的，而且使用连接的方式，性能要比嵌套查询高出很多。

（2）连接查询是关系数据库中最重要的查询。在关系数据库管理系统中，数据之间的关系不需要确定表的建立时间，实体的所有信息通常都存储在表中。检索数据时，通过连接操作查询存储在多个表中的不同实体的信息。当查询涉及多个关系时，用嵌套查询逐步求解结构层次清楚，易于构造具有结构化程序设计的优点，但是相比于连接运算，目前商用关系数据库管理系统对嵌套查询的优化做的还不够完善，在实际应用中，能够用连接运算表达的查询尽可能采用连接运算。

3. 统计函数和分组统计函数有什么不同？

答：统计函数主要有，COUNT()用于统计数据数量，SUM()用于统计数据的和AVG()统计平均值及MIN()得到最小值和MAX()得到最大值。

分组统计函数常在数组重复的时候使用，通过对统计函数的嵌套使用计算数值，分组筛选后再计算统计值，将数据按分组关键字分组，分组后聚集函数将作用于每一组，即每一组都有一个函数值。

4. 使用GROUP BY <条件条件>子句后，语句中的统计函数的运行结果有什么不同？

答：GROUP BY语法可以根据给定数据列的每个成员对查询结果进行分组统计；将 WHERE 子句与 GROUP BY 子句一起使用，分组查询可以在形成组和计算列函数之前具有消除非限定行的标准 WHERE 子句；在GROUP BY子句之后使用HAVING子句，可应用限定条件进行分组，以便系统仅对满足条件的组返回结果。

**实验三 数据库的维护实验**

**一、实验目的**

要求学生熟练使用和掌握数据库的维护操作，包含数据的插入、检索、修改，掌握SQL Server中用户、角色及操作权限的管理方法，学会创建和使用规则、缺省和触发器。

**二、实验内容**

1. 把全部红色零件颜色改为粉红色。

2. 由S1供给J1的零件P1今改为由S2供应，作必要修改。

3. 删去全部蓝色零件及相应的SPJ记录。

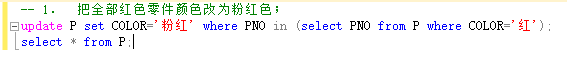
4. 把全部螺母的重量置为0。

5. 为SPJ表的QTY字段设计CHECK约束：0〈 QTY〈1000。

6. 实现对SPJ表的操作权限管理的使用。

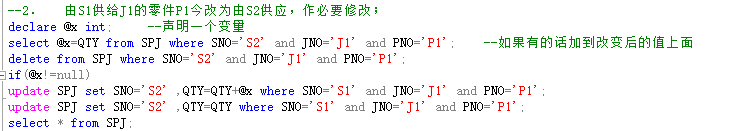
**三、实验实现及结果**

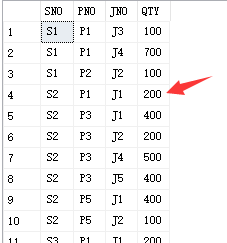
1. 把全部红色零件颜色改为粉红色；



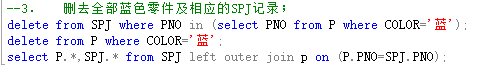


2. 由S1供给J1的零件P1今改为由S2供应，作必要修改；





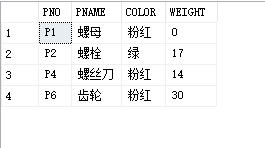
3. 删去全部蓝色零件及相应的SPJ记录；





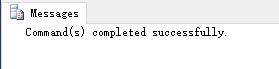
4. 把全部螺母的重量置为0；





5. 为SPJ表的QTY字段设计CHECK约束：0〈 QTY〈1000；





6. 实现对SPJ表的操作权限管理的使用。





**四、注意事项**

1. 数据更改时要注意外码约束。

2. 数据删除时要注意外码约束。

3. 用户、角色和权限的职能，以及它们之间的关系。

**五、思考题**

1. 为什么不能随意删除被参考表中的主码。

答：数据库中各个表之间由主码进行联系，删掉可能破坏实体完整性。

数据库中一般不允许更改主码数据。如果需要更改主码数据时，怎样处理？

答：先删除主键约束再重新添加。

2. 两种SQL Server的安全认证模式及特点。

答：Windows身份验证：这种形式的身份验证依赖于Windows来做验证身份。访问SQL Server对象的权限被分配给Windows登录名。这种类型的验证仅适用于SQL Server运行在支持Windows NT或Kerberos身份验证的Windows版本(Windows 2000及之后)。

3. SQL Server身份验证：SQL Server可以完全依靠自己进行验证。在这种情况下，你可以创建独特的用户名称(SQL Server称做登录名)和密码。用户或应用程序连接到SQL Server和访问所提供的凭据。权限直接给登录名或通过角色分配。

4. 什么是触发器？主要功能是什么？

答：触发器是SQL server 提供给程序员和数据分析员来保证数据完整性的一种方法，它是与表事件相关的特殊的存储过程，它的执行不是由程序调用，也不是手工启动，而是由事件来触发。

功能：

可在写入数据表前，强制检验或转换数据。

触发器发生错误时，异动的结果会被撤销。

部分数据库管理系统可以针对数据定义语言使用触发器，称为DDL触发器。

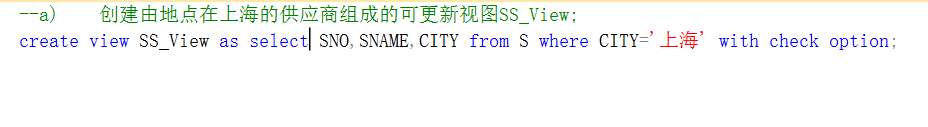
可依照特定的情况，替换异动的指令。

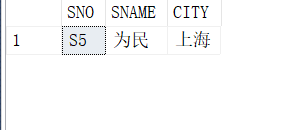
**六、补充题**

1） 针对给定的数据库模式，以及相应的应用需求，创建视图、创建带WITH CHECK OPTION的视图，并验证视图WITH CHECK OPTION选项的有效性。理解和掌握视图消解执行原理，掌握可更新视图和不可更新视图的区别

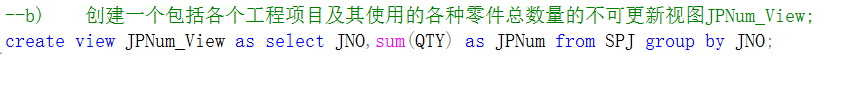
* 对视图的操作：

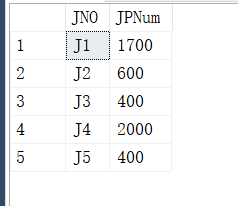
1. 创建由地点在上海的供应商组成的可更新视图SS\_View;



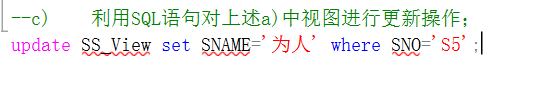


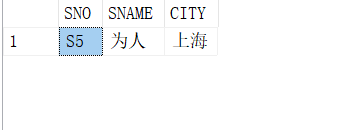
1. 创建一个包括各个工程项目及其使用的各种零件总数量的不可更新视图JPNum\_View;



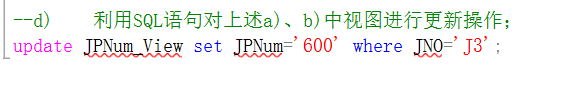


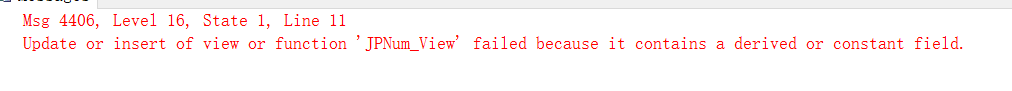
1. 利用SQL语句对上述a)中视图进行更新操作；





1. 利用SQL语句对上述b)中视图进行更新操作；

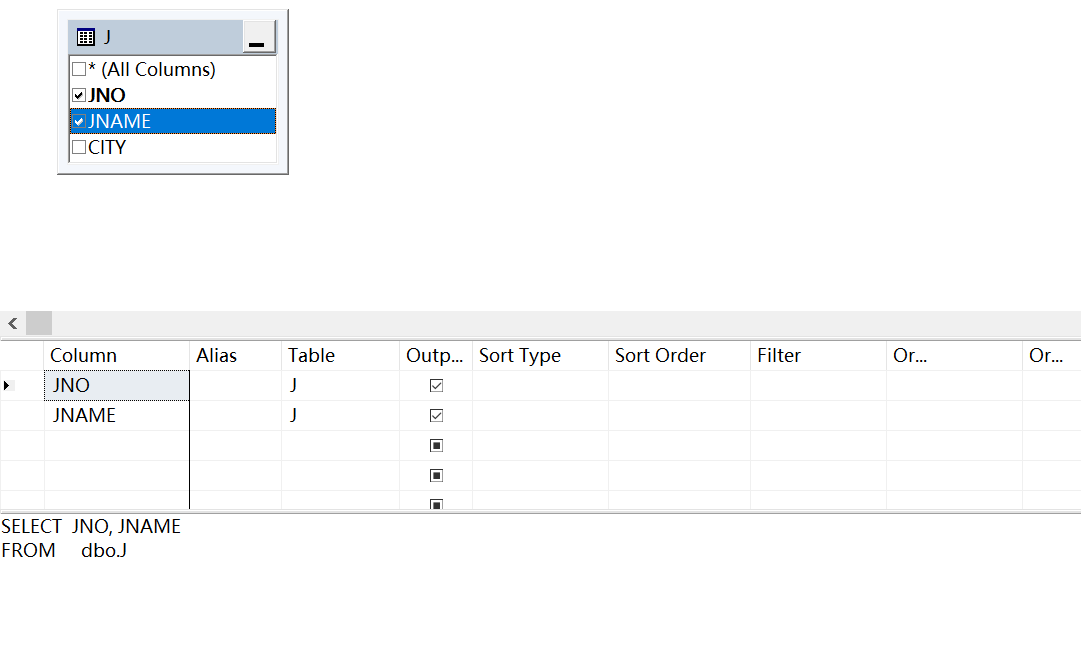


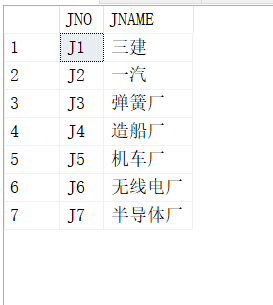


由于JPNum是由函数计算得出，所以不能更改。

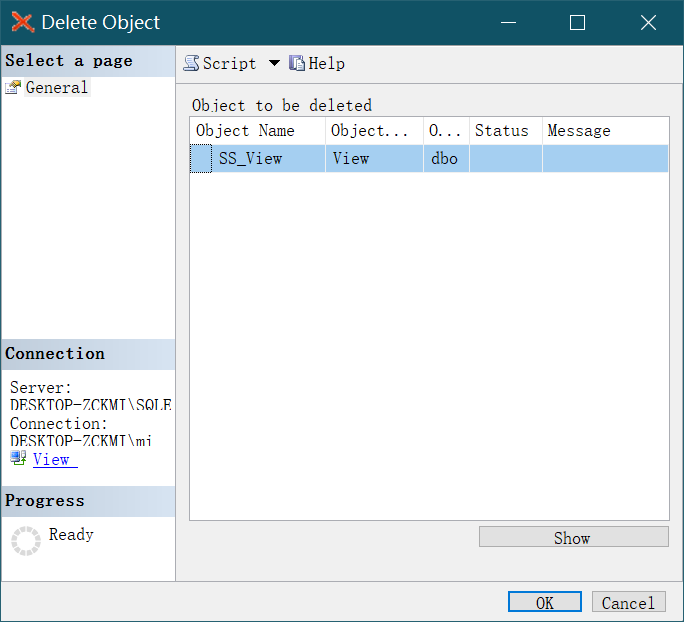
1. 利用Management Studio创建和删除视图。

新建：





删除：



**实验四 数据库的安全性、完整性实验**

**一、实验目的**

1. 掌握自主存取控制权限的定义和维护方法。

2. 掌握实体完整性的定义和维护方法。

3. 掌握参照完整性的定义和维护方法。

4. 掌握用户自定义完整性的定义和维护方法。

**二、实验内容**

1.定义用户、角色，分配权限给用户、角色，回收权限，以相应的用户名登录数据库验证权限分配是否正确。选择一个应用场景，使用自主存取控制机制设计权限分配。可以采用两种方案。方案一：采用sa超级用户登录数据库，完成所有权限分配工作，然后用相应用户名登陆数据库以验证权限分配正确性；方案二：采用sa用户登陆数据库创建三个部门经理用户，并分配相应的权限，然后分别用三个经理用户名登陆数据库，创建相应部门的USER, ROLE，并分配相应权限。验证权限分配之前，请备份好数据库；针对不同用户所具有的权限，分别设计相应的SQL语句加以验证。

2.定义实体完整性，删除实体完整性。能够写出两种方式定义实体完整性的SQL语句：创建表时定义实体完整性、创建表后定义实体完整性。设计SQL语句验证完整性约束是否起作用。

3.定义参照完整性，定义参照完整性的违约处理，删除参照完整性。写出两种方式定义参照完整性的SQL语句：创建表时定义参照完整性、创建表后定义参照完整性。

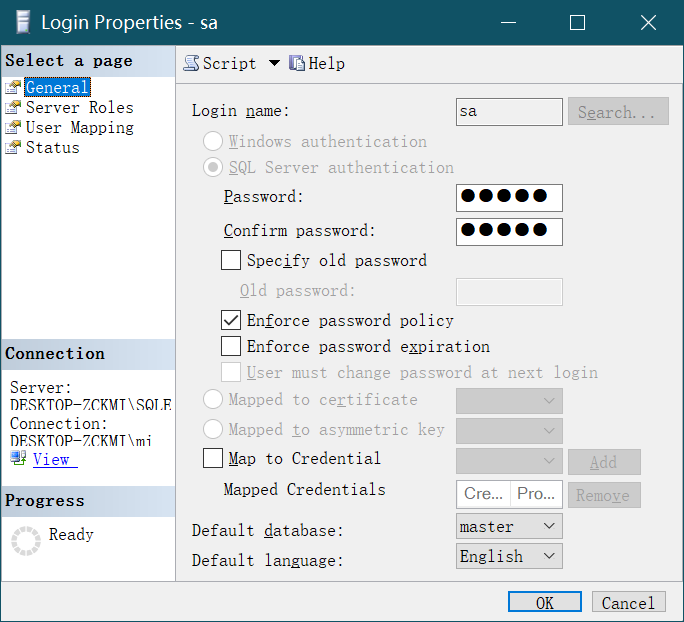
4.针对具体应用语义，选择NULL/NOT NULL、DEFAULT，UNIQUE、CHECK等，定义属性上的约束条件。

**三、实验实现及结果**

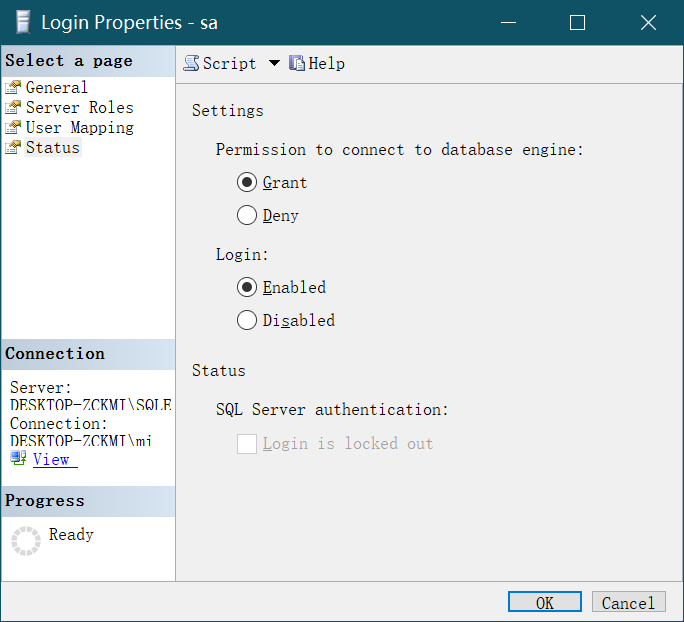
1.定义用户、角色，分配权限给用户、角色，回收权限，以相应的用户名登录数据库验证权限分配是否正确。选择一个应用场景，使用自主存取控制机制设计权限分配。可以采用两种方案。方案一：采用sa超级用户登录数据库，完成所有权限分配工作，然后用相应用户名登陆数据库以验证权限分配正确性；方案二：采用sa用户登陆数据库创建三个部门经理用户，并分配相应的权限，然后分别用三个经理用户名登陆数据库，创建相应部门的USER, ROLE，并分配相应权限。验证权限分配之前，请备份好数据库；针对不同用户所具有的权限，分别设计相应的SQL语句加以验证。

1. 首先开启sa超级用户

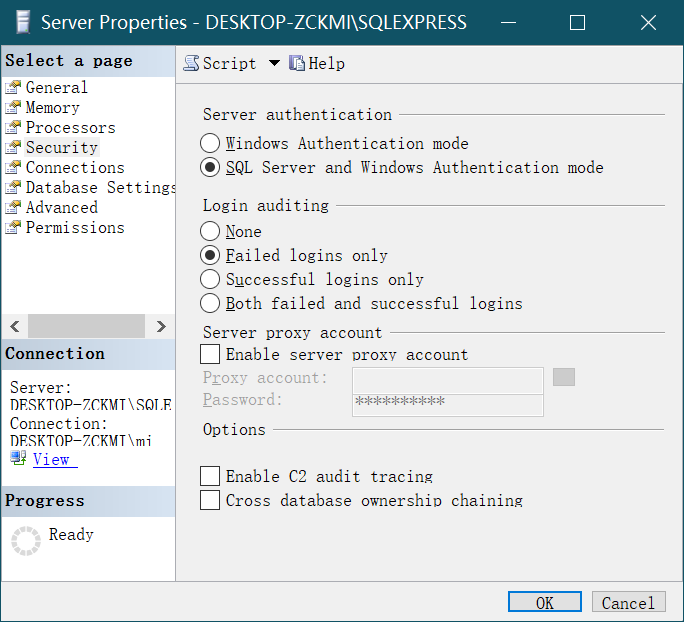
在Security中的logins中点击sa用户，如下显示：



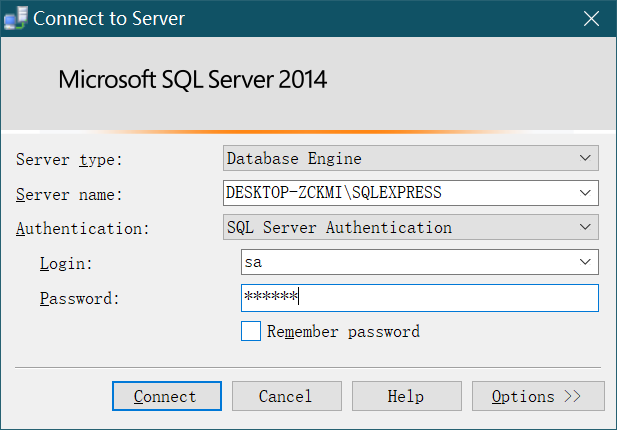
并在state中设置Enabled，如下显示：



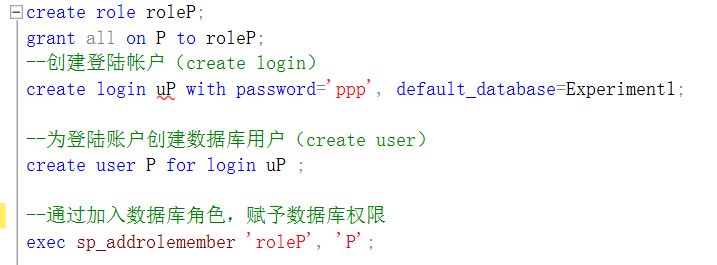
并在数据库的设置选项选定混合模式，如下显示：



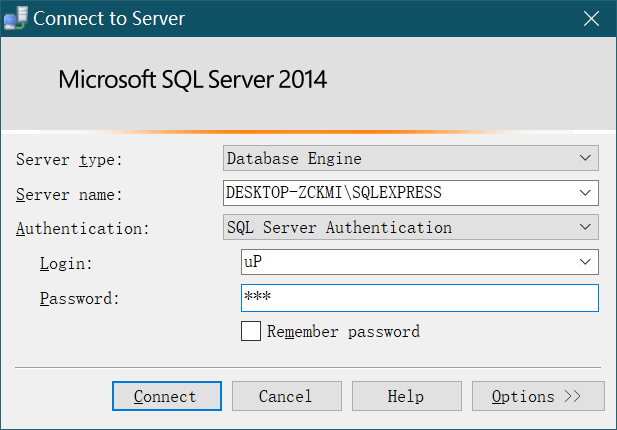
登录sa 超级用户账号，如下显示：



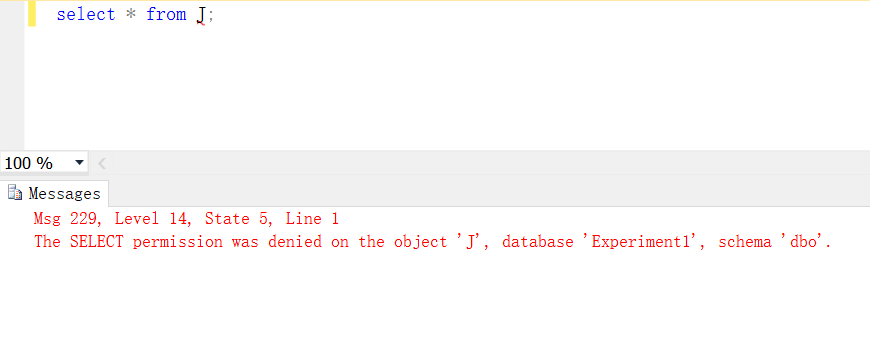
创建角色P，授权可以登录P表，其余表无权限，如下图所示：

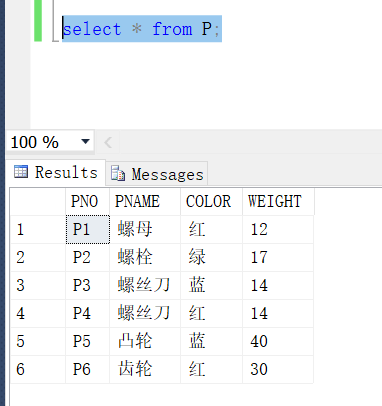


登录用户uP，如下所示：

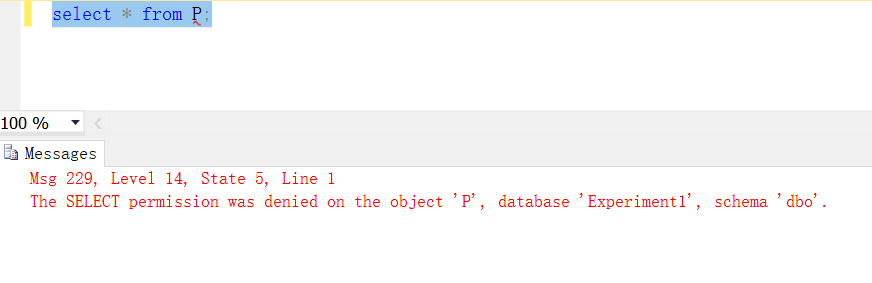


查询J表，被拒绝，如下所示：





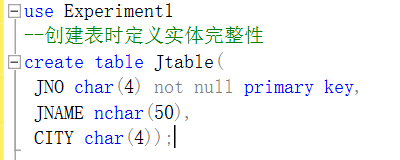
可见授权有效，同理创建用户J，授权J表，无法访问J表，如下所示：



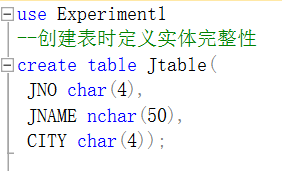


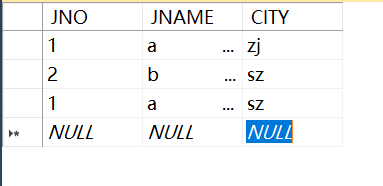
2.定义实体完整性，删除实体完整性。能够写出两种方式定义实体完整性的SQL语句：创建表时定义实体完整性、创建表后定义实体完整性。设计SQL语句验证完整性约束是否起作用。

创建表时定义实体完整性,在创建表时定义主键即可，如下所示：

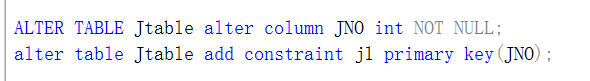


创建表后定义实体完整性，首先创建一个无约束的表，如下所示：

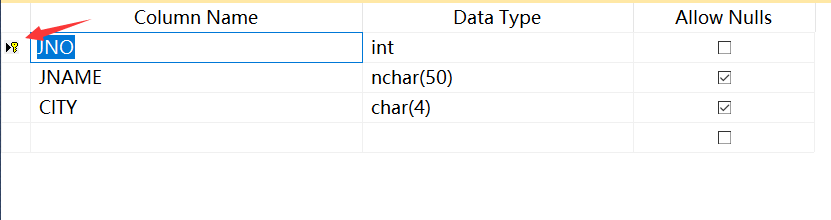




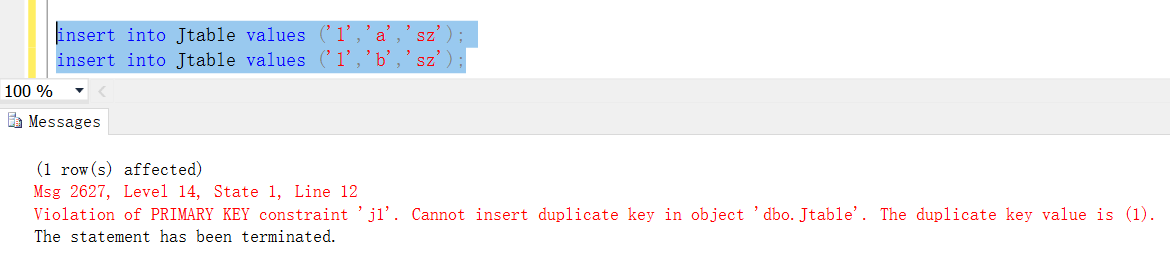
输入增加primary key语句报错，经查询发现需要设置not null条件，如下所示：



如下所示，可见添加完成：

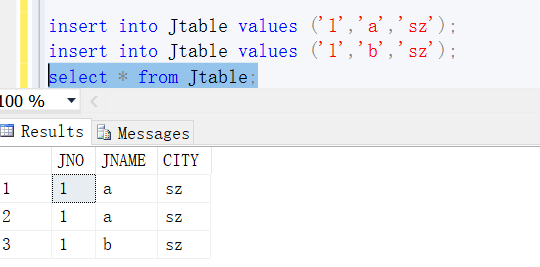


验证实体完整性语句如下，可见完成验证实体完整性：



删除实体完整性，操作如下：

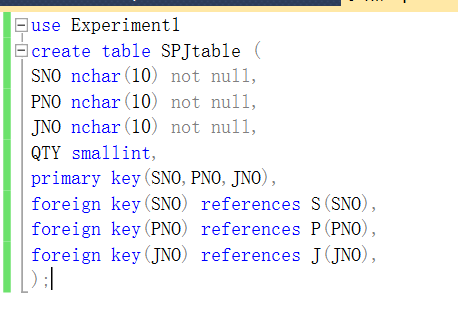




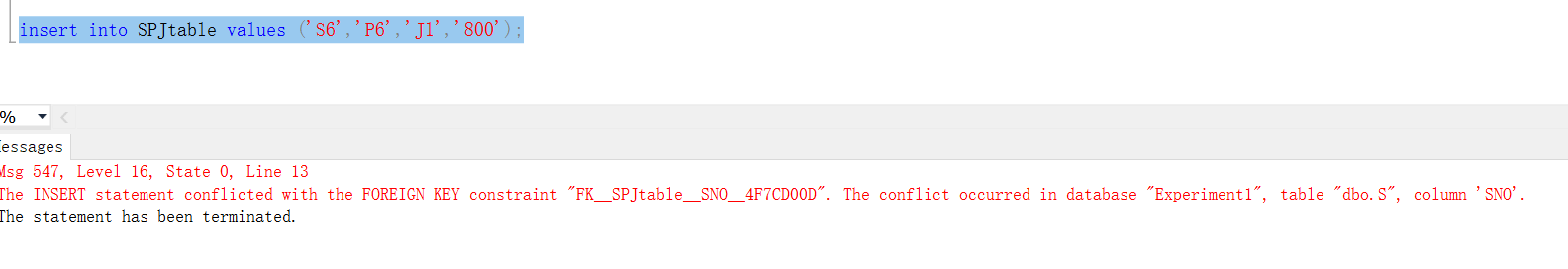
可见实体完整性已被删除。

3.定义参照完整性，定义参照完整性的违约处理，删除参照完整性。写出两种方式定义参照完整性的SQL语句：创建表时定义参照完整性、创建表后定义参照完整性。

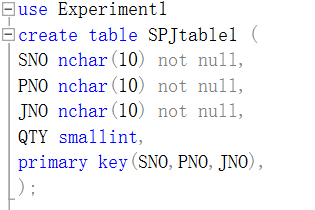
创建表时定义参照完整性，如下图所示：



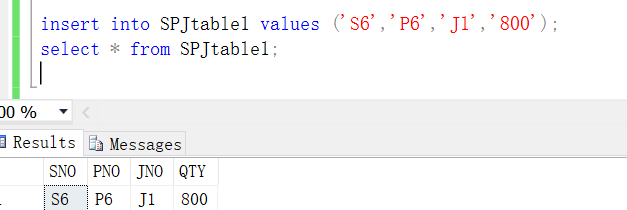
验证语句及警告如下，可见完成参照完整性：



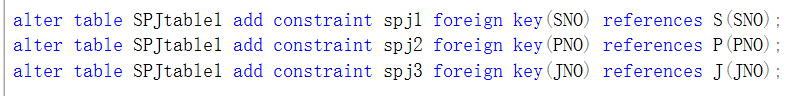
为在创建表后定义参照完整性，先创建表，如下所示：



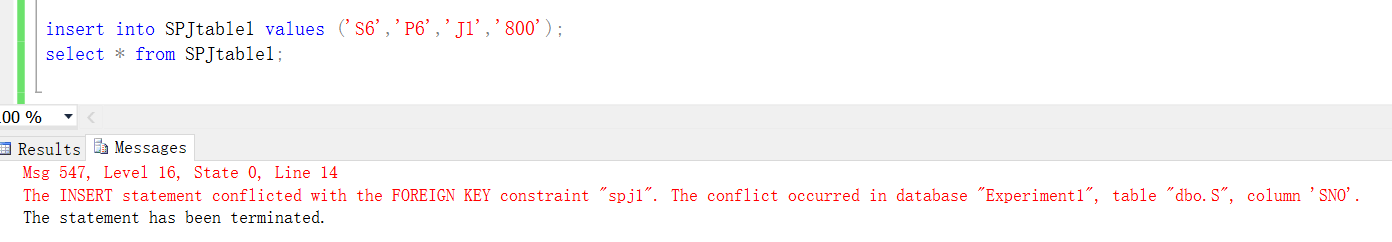
验证语句执行，证明没有参照完整性：



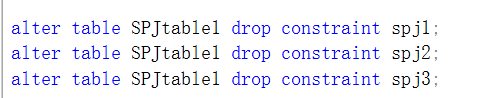
执行参照完整性语句，如下显示：



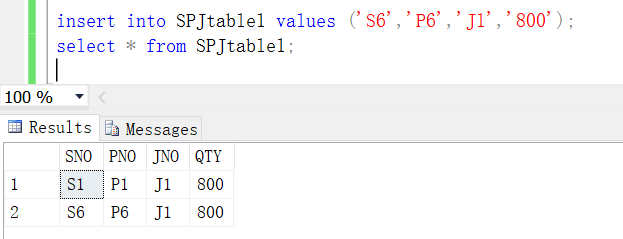
出现如下报错，可说明错误，完成验证。



执行删除约束语句，如下显示：

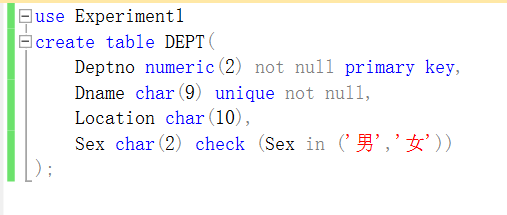


验证可见约束被删除。



4.针对具体应用语义，选择NULL/NOT NULL、DEFAULT，UNIQUE、CHECK等，定义属性上的约束条件。

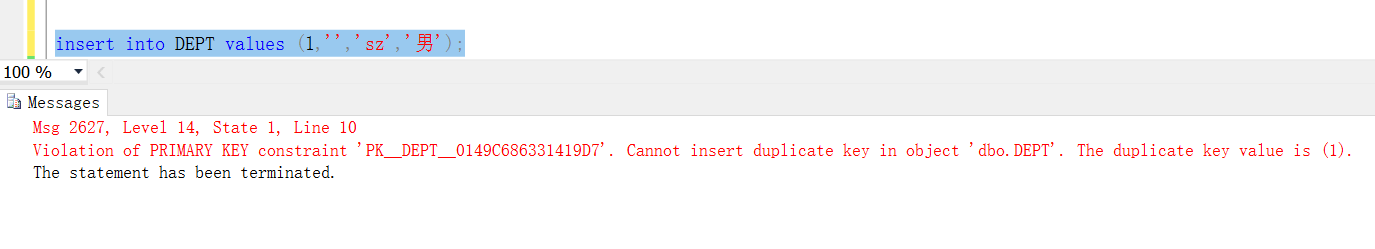
定义表部门，自定义了约束与属性，下面进行验证：



首先验证实体完整性，验证成功：



再次验证NOT NULL约束，验证成功：



最后验证CHECK约束，验证成功：

