**实验1 企业管理器的使用**

**试验一 注册服务器**

**目的：1**掌握利用企业管理器管理数据库服务器。

**一 启动数据库服务器**

打开服务管理器，“开始/继续“键为绿色。

**二 注册数据库服务器**

1 打开“企业管理器”

2 在企业管理器中的左边，选中 “Sql Server 组”。

3 如已经有一台服务器，则那台服务器就是你的数据库服务器。

选择“删除 sql server 注册”,删除该服务器。

4 然后，在“sql server组”上用Mouse右键弹出菜单，选择“新建注册”，

5 输入你的计算机的名字。

使用 sql server 身份验证，用户名：sa , 密码：空。

注意：如该服务器的验证模式为：仅 windows ,则需要更改为：混合模式。

三 更改验证模式

1 使用 window 身份验证模式注册服务器。

2 然后选中该台服务器，按右键，选择“属性”，再选“安全”

3 设置身份验证模式为混合模式：Sql server和windows

**试验二 创建数据库**

**目的：1**掌握利用企业管理器创建、维护数据库。了解sql server的数据库的存储结构。

**要求**：1 建立数据库

2 修改数据库

3 删除数据库

**一 建立school 数据库**

1 使用SQL-Server 的企业管理器创建数据库 school 。

要求记录：

1）数据库school 对应的数据库文件在那个文件夹。

2）数据库school 对应的数据库文件名。

2 删除School数据库

**二 建立 sale 数据库**

1 建立sale数据库,要求数据库存储在c:\data文件夹下，初始大小为5MB ,增量为 1MB。

2使用SQL-Server 的企业管理器，将数据库的每次增量改为20%。

思考：1 如何将数据库school从一台计算机部署到其他计算机上。

2 如何将数据库 sale 的文件存储在c盘和d盘。

**试验三 创建表**

**目的：**1掌握掌握利用企业管理器创建、维护表的方法。

2 了解SQL Server 的数据类型。

**要求：**1 建立表 2 修改表结构 3 删除表

**一 在数据库 school 中利用企业管理器建立student , sc,course 表。**

学生表、课程表、选课表属于数据库 School ，其各自的数据结构如下：

学生 Student (Sno,Sname,Ssex,Sage,Sdept)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 含义 | 数据类型 | 长度 |
| 1 | Sno | 学号 | 字符型(char) | 6 |
| 2 | Sname | 姓名 | 字符型(varchar) | 8 |
| 3 | Ssex | 性别 | 字符型(char) | 2 |
| 4 | Sage | 年龄 | 整数 (smallint) |  |
| 5 | sdept | 系科 | 字符型(varchar) | 15 |

课程表 course(Cno,Cname,Cpno,Ccredit)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 含义 | 数据类型 | 长度 |
| 1 | Cno | 课程号 | 字符型(char) | 4 |
| 2 | cname | 课程名 | 字符型(varchar) | 20 |
| 3 | Cpno | 先修课 | 字符型(char) | 4 |
| 4 | Ccredit | 学分 | 短整数 (tinyint) |  |

学生选课 SC(Sno,Cno,Grade)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 含义 | 数据类型 | 长度 |
| 1 | Sno | 学号 | 字符型(char) | 4 |
| 2 | Cno | 课程名 | 字符型(char) | 6 |
| 3 | Grade | 成绩 | 小数(decimal) | 12,2 |

**二 设定主码**

1 Student表的主码：sno

2 Course表的主码：cno

3 Sc表的主码：sno，cno

**三 表结构修改**

1 在student 表中添加列：

家庭地址 address 长度为 60 varchar 型

入学日期 inDate 日期型

注意：设定后保存。

2 将家庭地址 address 长度为 50

3 删除 student 表的 inDate 列

4 删除表 sc 的主码

**四 删除表**

1 删除表 SC

**五 设定约束**

1 student表中的 姓名不可为空

2 student 表中的系科不可为空，默认值为 ‘JSJ’

思考：

1 student表的性别不能为空且取值范围为{男，女} ，年龄必须大于16岁

**试验四 数据输入**

**目的：**1掌握利用企业管理器进行数据处理的方法。

**要求：**1 数据数据 2 修改数据 3 删除记录

**一 输入数据**

1 student 表数据输入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学号 | 姓名 | 性别 | 年龄 | 系科 |
| 5001 | 赵强 | 男 | 20 | SX |
| 5002 | 杨丽华 | 女 | 21 | JSJ |
| 5001 | 李静 | 女 | 22 | SX |
|  |  |  |  |  |

记录输入过程中遇到的问题。

2 course 表数据的输入

|  |  |  |  |
| --- | --- | --- | --- |
| Cno | Cname | Cpno | Ccredit |
| 1801 | C语言 |  | 4 |
| 1802 | 数据结构 | 1081 | 4 |

3 SC 表数据的输入

|  |  |  |
| --- | --- | --- |
| Sno | Cno | Grade |
| 5001 | 1801 | 90 |
| 5001 | 1802 | 79 |
| 5002 | 1801 | 91 |
| 5008 | 1801 | 99 |
|  |  |  |

记录输入过程中遇到的问题。

**二 修改数据**

1 把 5001 学生的年龄改为 22岁

2 把 5001学生的1802课程的成绩为该81分

**三 删除记录**

1 把 5001 学生的年龄改为 22岁

2 把 sc表中 5001学生的1802课程的记录删除

思考：1 sc表中5008 这一条记录的存在是否合理？原因？

如何可以避免此种情况发生。

**试验五 登录到数据库服务器**

**目的：**1了解如何登录到另外一台数据库服务器。

**一 察看其他电脑的服务器名**

1 打开服务管理器，察看你需要登录的数据库服务器的服务器名。

2 登录服务器

1）在企业管理器中的左边，选中 “Sql Server 组”，Mouse右键弹出菜单，选择“新建注册”，

2）输入你想要登录的服务器名字。

3）使用 sql server 身份验证，用户名：sa , 密码：空。

注意：如该服务器的验证模式为：仅 windows ,则需要更改为：混合模式。

3 注册成功后

1）进入该服务器的 school 数据库。

2）打开表 student 表，把0001学生的年龄改为24岁。

3）再在那台服务器上查询0001学生的年龄。

**实验2 SQL语言（试验一-试验三）**

**试验一 数据库创建**

**目的：**1掌握利用SQL语言进行数据库的创建、维护。

2 sp\_helpdb 命令

**要求：**1 创建数据库 2 修改数据库 3 删除数据库

**一 建立school 数据库**

1 使用查询分析器创建数据库 school

2 使用 SP\_helpdb 查询数据库 School 的信息

3 使用SQL-Server 的企业管理器查看数据库 school 的信息。

4 记录：

1）school 数据库文件所在的文件夹。

2）school 数据库的文件名

**二 删除School数据库**

1 使用查询分析器删除数据库 school

2 使用SQL-Server 的企业管理器删除数据库 school 。

**三 create Database 深入研究**

1 建立school数据库,要求数据库存储在c:\data文件夹下，初始大小为5MB ,增量为 1MB。

2使用SQL-Server 的企业管理器，将数据库的每次增量改为20%。

**试验二 创建表**

**目的：**1 掌握利用SQL语言创建表的方法。

2 sp\_help 命令

**要求：**1 创建表 2 修改表结构 3 删除表

**一 写出使用 Create Table 语句创建表 student , sc,course 的SQL语句。**

学生表、课程表、选课表属于数据库 School ，其各自得数据结构如下：

学生 Student (Sno,Sname,Ssex,Sage,Sdept)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 含义 | 数据类型 | 长度 |
| 1 | Sno | 学号 | 字符型(char) | 6 |
| 2 | Sname | 姓名 | 字符型(varchar) | 8 |
| 3 | Ssex | 性别 | 字符型(char) | 2 |
| 4 | Sage | 年龄 | 整数 (smallint) |  |
| 5 | sdept | 系科 | 字符型(varchar) | 15 |

课程表 course(Cno,Cname,Cpno,Ccredit)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 含义 | 数据类型 | 长度 |
| 1 | Cno | 课程号 | 字符型(char) | 4 |
| 2 | cname | 课程名 | 字符型(varchar) | 20 |
| 3 | Cpno | 先修课 | 字符型(char) | 4 |
| 4 | Ccredit | 学分 | 短整数 (tinyint) |  |

学生选课 SC(Sno,Cno,Grade)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 含义 | 数据类型 | 长度 |
| 1 | Sno | 学号 | 字符型(char) | 4 |
| 2 | Cno | 课程名 | 字符型(char) | 6 |
| 3 | Grade | 成绩 | 小数(decimal) | 12,1 |

**二 把创建表的sql 语句的脚本存储到文件 school.sql 。**

**三 使用 SP\_HELP 查看表 student 的表结构**

**利用企业管理器查看表 sc 的表结构**

**四 利用 sql 语句表结构修改**

1 在student 表中添加列：

家庭地址 address 长度为 60 varchar 型

入学日期 inDate 日期型

完成后用sp\_help 查看是否成功。

2 将家庭地址 address 长度为 50

完成后用sp\_help 查看是否成功。

3 删除 student 表的 inDate 列

**五 删除表**

1 删除表 sc

2 删除表 student

3 删除表 course

**试验三 创建数据完整性**

**目的：**1掌握创建数据完整性约束的命令。

2 掌握完整性约束的修改、删除。

**要求：**1 能建立完整性约束 2 修改完整性约束 3 删除完整性约束

**一 写出带有完整性约束的 Create Table 命令建立表student、course、sc 。要求：**

1 Student表的主码：sno

student 的约束：

* 姓名不可为空，且唯一
* 性别 不能为空且取值范围为{男，女}
* 年龄大于16岁
* sdept 默认为 ‘JSJ’ 系

2Course表的主码：cno

course 的约束：

* Ccredit 取值范围{ 0 ,1,2,3,4,5 }
* 课程表的每一行的 Cno 与 cpno 不可相同

3 Sc表的主码：sno，cno 。主码名为 PK\_SC

Sc的外码：

* 外码：SC 表的sno 参照表 student 的 sno
* 外码：sc 表的Cno 参照表 course 的 cno

4 把上述创建表的sql 语句的脚本存储到文件 createSchool.sql 。

**二 使用 SP\_HELP 查看表 student 的主码名，约束名，并记录。**

使用 SP\_HELP 查看表 sc 的主码名，外码名，并记录。

**三 利用alter table 添加、删除完整性约束**

1 删除 SC 的主码，sc表的主码名为 pk\_sc

复习在 Sql-Server 企业管理器中如何完成。

2 删除 SC 表**参照 course 表**的外码。

如何知道 SC表**参照 course 表**的外码的名字。

3 添加 SC 表的主码。主码名为 PK\_SC

4 添加 SC 表的Cno 的外码 ，参照表Course的Cno.

5加自定义约束：表 SC的成绩只能在 0 – 100分之间。

**四 使用 Sql-Server 企业管理器完成：**

1 删除 SC 表**参照 course 表**的外码。

2 建立 SC 表的Cno 的外码 ，参照表Course的Cno.

\*使该外码具有级联修改的功能。

3 删除表 SC的成绩只能在 0 – 100分之间的约束.

3加自定义约束：表 SC的成绩只能在 0 – 100分之间。

**五 使用**

select \* from student 查看信息

select \* from course 查看信息

select \* from sc 查看信息

**实验3 SQL语言（试验四-试验六）**

**试验四 数据完整性试验**

**目的：**1理解实体完整性、参照完整性、用户自定义完整性的作用

2 特别掌握外码的作用。

**要求：**记录试验中遇到的问题，并写出原因。

实验前需要利用试验三完成的脚本文件 createSchool.sql，重新建立数据库 school 。

**一 实体完整性**

1 student 表数据输入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学号 | 姓名 | 性别 | 年龄 | 系科 |
| 3001 | 赵达 | 男 | 20 | SX |
| 3002 | 杨丽 | 女 | 21 | JSJ |
| 3001 | 李寅 | 女 | 21 | SX |
|  |  |  |  |  |

* 输入上述数据，记录出现的问题，说明原因。
* select \* from student 查看你输入了几行数据。

2 course 表数据的输入

|  |  |  |  |
| --- | --- | --- | --- |
| Cno | Cname | Cpno | Ccredit |
| 1081 | 电子商务 |  | 4 |

3 SC 表数据的输入

|  |  |  |
| --- | --- | --- |
| Sno | Cno | Grade |
| 3001 | 1081 | 90 |
| 3001 | 1081 | 79 |
|  |  |  |

输入上述数据，记录出现的问题，说明原因。

**二 用户自定义完整性约束**

表student 有用户自定义约束：

**性别 不能为空且取值范围为{男，女}**

**年龄大于16岁**

表course 的自定义约束：

**Ccredit 取值范围{ 0 ,1,2,3,4,5 }**

**课程表的每一行的 Cno 与 cpno 不可相同**

1 student 表数据输入

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学号 | 姓名 | 性别 | 年龄 | 系科 |
| 3005 | 赵达 | 男 | 14 | SX |
| 3006 | 杨丽 | 南 | 21 | JSJ |

* 输入上述数据，记录出现的问题，说明原因。
* select \* from student 查看你输入了那些数据。

2 course 表数据的输入

|  |  |  |  |
| --- | --- | --- | --- |
| Cno | Cname | Cpno | Ccredit |
| 1085 | C++ |  | 9 |
| 1086 | 语文 | 1086 | 3 |

* 输入上述数据，记录出现的问题，说明原因。
* select \* from student 查看你输入了那些数据。

3 SC 表数据的输入

|  |  |  |
| --- | --- | --- |
| Sno | Cno | Grade |
| 3002 | 1081 | 128 |
|  |  |  |

* 输入上述数据，记录出现的问题，说明原因。
* select \* from student 查看你输入了那些数据。

**三 参照完整性约束**

* 掌握表之间建立外码后，对被参照表的如下操作会有何影响：

修改主码、插入新行、删除新行？

* 对参照表添加新行、删除行、修改外码值有何影响？
* 掌握级联修改、级联删除的概念。

**注意：**

**表SC的 Sno是外码，参照student的sno。**

**表SC的 Cno是外码，参照course的cno。**

**1 输入实验前的数据**

学生表 Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sno | Sname | Ssex | Sage | Sdept |
| 4001 | 赵尹 | 男 | 20 | SX |
| 4002 | 杨开 | 女 | 20 | JSJ |

课程表 course

|  |  |  |  |
| --- | --- | --- | --- |
| Cno | Cname | Cpno | Ccredit |
| 1088 | Java |  | 5 |
| 1089 | 数学 |  | 3 |

学生选课 SC

|  |  |  |
| --- | --- | --- |
| Sno | Cno | grade |
| 4001 | 1088 | 90 |
| 4002 | 1088 | 86 |
|  |  |  |

**2 试验过程**

**1) 在SC表中添加新行：**

|  |  |  |
| --- | --- | --- |
| Sno | Cno | Grade |
| 4001 | 1066 | 76 |

记录试验结果.,写出出现此结果的原因.

**2) 在student表中添加新行**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sno | Sname | Ssex | Sage | Sdept |
| 4003 | 赵辉 | 男 | 21 | SX |

记录试验结果.,写出出现此结果的原因.

**3) 删除student 表的 4001 ，4002学生**

记录试验结果.,写出出现此结果的原因.

思考：

* 删除SC表的记录有限制吗？
* 采取什么技术能使不能成功执行的命令变得可以执行，且使数据库保持数据完整性。

**4) 把 student 表的学号 4003 改为 4018 , 4001改为4021。**

记录试验结果.,写出出现此结果的原因.

思考：采取什么技术能使本题不能执行的命令可以执行，且使数据库保持数据完整性。

**5) 把sc表中的如下记录的学号从4001改为4011。**

|  |  |  |
| --- | --- | --- |
| Sno | Cno | Grade |
| 4001 | 1088 | 90 |

记录试验结果.,写出出现此结果的原因.

* 如不成功，则可以采取什么方法来实现此要求。
* 如不成功，那么把4001修改为4003，能成功吗？

思考: 参照完整性规则中,外码可以为空, 但SC表中的外码可以为空吗?为什么?

举一个外码可以为空的例子。

**试验五 索引**

**目的：**掌握索引的建立、删除的方法。

一 创建索引

1 建 student 的索引

为姓名建立索引，索引名：Ix\_student\_sname

为系科建立索引，索引名：Ix\_student\_sdept

2 SC 的索引

为课程号建立索引： ix\_sc\_cno

3 Course 的索引

为课程名建立唯一性索引 ：Ix\_course\_cname

4 如何 SP\_HELP 查看索引刚才建立的索引？

如何在企业管理器中查看索引？

二 删除索引 course 表的索引 IX\_course\_cname

三 思考：如何把索引 IX\_student\_sname 修改为唯一性索引？

\*四 思考建立索引的目的

1 输入下列存储过程，该程序生成大量数据供测试：

create procedure usp\_makedata as

declare @nCnt int , @sNo varchar(6) , @sname varchar(8)

set @nCnt =12000 --计数器

while @nCnt<999999

begin

set @nCnt = @nCnt + 1

set @sNo = convert(varchar(6) ,@nCnt)

set @sName = '张'+@sno

insert into student (sno,sname,ssex,sage) values ( @sno,@sname,'男',20)

end

return

2 exec usp\_makedata --生成测试数据

3 输入下述测试程序：

create procedure usp\_test as

declare @nCount int ,@data int

set @nCount=0

while @nCount<100

begin

select @data=count(\*) from student where sname <'张3800' or sname>'张8800'

set @nCount =@nCount + 1

end

4 测试

1）建立姓名的索引，查看运行时间(8秒).

create index ix\_student\_sname on student(sname) --建立索引

exec usp\_test

2) 删除姓名索引，查看运行时间（2分11秒），比较与1）的时间长短。

drop index student.ix\_student\_sname --删除索引

exec usp\_test

**试验六 更新数据**

**目的：**掌握insert,update ,delete 语句的使用。

**一 insert**

1 写出把下述学生的信息添加到student表中的命令。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学号 | 姓名 | 性别 | 年龄 | 系科 |
| 4001 | 赵茵 | 男 | 20 | SX |
| 4002 | 杨华 | 女 | 21 |  |

**2 批量插入数据**

1） 建立一个新表 sc\_name ，有属性 sno , sname , ssex , cno , grade 。

2） 把 SX 系学生的sno,sname,ssex, cno , grade 插入到表 sc\_name 中。

3） 察看 sc\_name 表的数据

**二 Update**

1 修改 0001 学生的系科为: JSJ

2 把陈小明的年龄加1岁，性别改为女。

2 修改李文庆的1001课程的成绩为 93 分

3 把“数据库原理”课的成绩减去1分

**三 Delete**

1 删除所有 JSJ 系的男生

2 删除“数据库原理”的课的选课纪录

思考：修改数据的命令与修改表结构的命令有何区别？

**实验4 SQL语言（试验七）**

**试验七 Sql 查询语句**

**目的: 掌握 Select 查询语句。**

**一 单表**

1查询年龄在19至21岁之间的女生的学号,姓名,年龄,按年龄从大到小排列。

2查询姓名中第2个字为“明”字的学生学号、性别。

3查询 1001课程没有成绩的学生学号、课程号

4查询JSJ 、SX、WL 系的年龄大于25岁的学生学号,姓名，结果按系及学号排列

5按10分制查询学生的sno,cno,10分制成绩

（1-10分 为1 ，11-20分为2 ，30-39分为3，。。。90-100为10）

6查询 student 表中的学生共分布在那几个系中。（distinct）

7查询0001号学生1001，1002课程的成绩。

**二 统计**

1查询姓名中有“明”字的学生人数。

2计算‘JSJ’系的平均年龄及最大年龄。

3查询学生中姓名为张明、赵英的人数

4计算每一门课的总分、平均分，最高分、最低分，按**平均分**由高到低排列

5 计算 1001,1002 课程的平均分。

6 查询平均分大于80分的学生学号及平均分

7 统计选修课程超过 2 门的学生学号

8 统计有10位成绩大于85分以上的课程号。

9 统计平均分不及格的学生学号

10 统计有大于两门课不及格的学生学号

**三 连接**

1查询 JSJ 系的学生选修的课程号

2查询选修1002 课程的学生的学生姓名 (不用嵌套及嵌套2种方法）

3查询数据库原理不及格的学生学号及成绩

4查询选修“数据库原理”课且成绩 80 以上的学生姓名(不用嵌套及嵌套2种方法）

5查询平均分不及格的学生的学号，姓名,平均分。

6查询女学生平均分高于75分的学生姓名。

7查询男学生学号、姓名、课程号、成绩。(一门课程也没有选修的男学生也要列出，不能遗漏)

**四 嵌套、相关及其他**

1 查询平均分不及格的学生人数

2 查询没有选修1002 课程的学生的学生姓名

3 查询平均分最高的学生学号及平均分 （2种方法 TOP , any , all）

\*4 查询没有选修1001，1002课程的学生姓名。

5 查询1002课程第一名的学生学号（2种方法）

6 查询平均分前三名的学生学号

7 查询 JSJ 系的学生与年龄不大于19岁的学生的差集

8 查询1001号课程大于90分的学生学号、姓名及平均分大于85分的学生学号、姓名

9 查询每门课程成绩都高于该门课程平均分的学生学号

10 查询大于本系科平均年龄的学生姓名

**实验5 SQL语言（试验八-试验十）**

**试验八 视图**

**目的: 掌握视图的建立、使用。**

1建立学生学号、姓名、性别、课程号、成绩的视图 v\_sc

查看V\_sc中的数据。

2 建立学生学号、姓名、出生年份的视图 v\_age

查看V\_age中的数据。

3 建立 ‘JSJ’ 系的学生学号、姓名、性别、年龄的视图 V\_JSJ

4 建立每门课程的平均分的视图 V\_avggrade

5 将 视图 v\_jsj 中 李文庆 的年龄改为21岁

6 察看 student 中李文庆的年龄

查看 v\_age 中李文庆的出生年月

7 查询每门课程的及格率

思考:

1 利用 V\_JSJ 视图,可以更新SX 的学生的年龄吗? 写出理由

如: update v\_jsj set sage=25 where sno=’ 0004’

0004 号学生为 SX 系.

2 create view v\_student (sno,sname,ssex,sage,sdept) as

select sno,sname,ssex,sage,sdept from student

学籍管理系统使用此视图。现把student 的列名 sname 改为 xm,请问原来的学籍管理系统能正常运行吗？如不能，则如何处理能使系统正常运行。

**试验九 安全性控制实验**

**目的:掌握Sql-server 的授权机制.**

1）建立新用户 mary , 密码1234

2) 授予 mary 可以访问 School 数据库的权力

3) 以mary 登录 sql-server ,

执行 select \* from student ,记录执行结果，说明原因。

4）将 course 的查询、更改权限授予 mary

5）把查询 student 表和修改学生学号的权限授予用户 mary,且他能将此权限转授他人。

6) 把对 course 表的更改权限从mary 收回

7) 把第5）小题授予mary的权限收回。

8）mary 只能查询 ‘1001’ 号课程的学生成绩,请问如何授权

思考: 1 sp\_addlogin , sp\_grantdbaccess 语句的区别.

2 如有200个人需要授权，SQL-SERVER如何简化授权机制。

**试验十 存储过程**

**目的: 掌握存储过程的概念、编程及使用**

1 编写一个存储过程 usp\_avgage , 向客户端返回每个系科的学生平均年龄。

系科 平均年龄

JSJ 21

SX 20

。。。

1) 编写存储过程的代码

2）调试、运行该存储过程。

2编写一个存储过程 usp\_sdept, 传入一个系科代码，返回该系的平均年龄，人数

3 编写存储过程 usp\_updateGrade , 传入参数为课程号,处理逻辑:

对传入的这门课,进行如下处理:

如某学生该门课成绩>80 , 则加 2 分

如某学生该门课成绩>60 , 则加 1 分

如某学生该门课成绩<=60 ,则减 1分

并且返回此门课的每个学生的最新成绩: 学号 成绩.

4 编写存储过程 usp\_g1 , 传入参数课程号，对该门课程进行如下处理，低于平均分5分不加分，低于平均分0-5分的加1分，高于平均分0-5加1分，高于平均分5分的加2分。

5 编写存储过程 usp\_comp\_age , 比较0001，0002学生的年龄的高低，输出： XXXX学生的年龄大

注意： XXXX为学生的姓名

6 编写存储过程 usp\_comp , 比较1001，1002课程的平均分的高低，输出： XXXX课程的平均分高

7 编写存储过程 usp\_comp\_age1 , 比较两个学生的年龄的高低，两个学生的学号有参数输入，最后输出： XXXX学生的年龄大。

注意： XXXX为学生的姓名

8 利用第8题的存储过程，判断0002，0003学生的年龄大小。

9 编写存储过程 usp\_comp1 , 传入两参数，课程号1，课程号2；比较这两门课的平均分的高低，输出： XXXX课程的平均分高

10 编写存储过程 usp\_comp\_age2 , 比较两个学生的年龄的高低，两个学生的学号有参数输入，最后把年龄大的学生的姓名、性别返回客户端。

11 编写存储过程 usp\_comp2 , 传入两参数，课程号1，课程号2；比较这两门课的平均分的高低，最后把平均分高的课程的课程名返回客户端。

12 编写存储过程 usp\_t1，传入参数为学号,把该学号的课程1001的成绩减到58分。每次只能减1分，用循环完成。

13 编写存储过程 usp\_t2，传入参数为系科，把该系科的学生每次加一岁，只要该系科有一个人的年龄达到28岁，即停止循环。每次只能减加1岁分，用循环完成。

15 编写存储过程 usp\_disp , 传入参数为课程号,处理逻辑: 返回每个学生的成绩等级。

成绩>=90 为优, 成绩>=80为良,成绩>=70 为中,成绩>=60为及格 ,成绩<=60为不及格。

返回结果如下：

学号 课程号 成绩 等第

0001 1001 91 优

0001 1002 78 中

……………..

16 编写一个存储过程，传入参数为学号，执行后，把该学号的学生按如下格式输出成绩：

（注意：只有一行）

学号 姓名 1001课程 1002课程 1003 课程 平均分

17 编写一个存储过程，传入参数为 系科，执行后，把该系科的学生按如下格式输出学生成绩：

学号 姓名 1001 课程 1002课程 1003 课程 平均分

18 编写存储过程，统计男女生1001，1002，1003各自的选修人数，输出格式如下：

性别 1001人数 1002人数 1003人数 小计

男 3 5 2 10

女 2 4 1 7

合计 5 9 3 17

（数据为示意数据）

19 编写一个存储过程，利用存储过程的参数返回数据库服务器上的日期时间。

思考：何时需要存储过程？

**实验6 SQL语言（试验十一-试验十四）**

**试验十一 触发器**

**目的: 了解触发器的机制及编程设计、使用**

一 建立学生表的触发器 usp\_addstudent，当增加学生时，SX系的学生不能超过30岁。

1 写出触发器

2 执行下列语句块：

begin tran

insert into student (sno,sname,ssex,sage,sdept)

values (‘0701’,’刘欢’,’男’,26,’SX’)

if @@error=0

commit

else

rollback

end

观察该学生是否加入到 student

3执行下列语句块：

begin tran

insert into student (sno,sname,ssex,sage,sdept) values (‘0702’,’赵欢’,’男’,31,’SX’)

if @@error=0

commit

else

rollback

end

观察该学生是否加入到 student

二 实现下列触发器

1 不能删除年龄大于25岁的学生记录。

2 建立触发器 usp\_delcourse , 使课程表中1001，1002，1003 三门课不会被删除。

注意如何调试。

3 对学生表建立一触发器，使更改后的年龄只能比原值大

4对sc表建立触发器，使‘JSJ’系的学生不可选择 ‘1004’号课程

5 对表 course 建触发器，实现级联删除的功能，但某课选修人数大于3则不能删除。

(先删除 sc 表对course 的外码)

\*三 建立一个触发器，使对sc表成绩的修改自动记录修改日志。

日志文件表(tablog)记录如下：

用户名 学号 课程号 原成绩 修改后成绩 更改日期

四 在School数据库中建立一个试验用的发票表bill，然后为发票bill建立触发器 utr\_money ,实现当输入单价和数量后，自动填写金额，即发票金额不输入，由单价、数量相乘后自动填写到金额中。

Create table bill(

billID char(8), --发票编号

date datetime, --开票日期

product char(10), --产品编号

price int , --单价

qty int , --数量

charge int , --金额

primary key (billid) )

思考: 触发器中 inserted , deleted 表的作用? 在触发器中如没有用到此两个表中的任何一个,你认为触发器还有意义吗?

**试验十二 恢复技术**

**目的：**1 掌握数据库的备份及恢复的方法。

2 了解备份方案的设定

一 完全备份的建立与恢复

1建立完全备份

USE school

GO

BACKUP DATABASE school TO DISK=’C:\schooldata.bak’

2查看备份文件中的信息

RESTORE FILELISTONLY FROM DISK=’c:\schooldata.bak’

RESTORE HEADERONLY FROM DISK=’c:\schooldata.bak’

3恢复完全备份

1) 先删除数据库 School

USE Master

GO

DROP DATABASE school

2) 然后恢复.

RESTORE DATABASE school from DISK=’c:\schooldata.bak’

3): 查看 school 的student 中的数据

二 建立差异备份

1 建立备份

1） 制作数据文件备份 schoolDiff.bak

2） 把学号 7001， 姓名：王海，性别：男，年龄为23 的学生加入student

3） 制作school 的差异备份 ，存入schoolDiff.bak

BACKUP DATABASE school TO DISK=’schoolDiff.bak’ WITH DIFFERENTIAL

4) 把学号 7002， 姓名：赵燕，性别：女,年龄为22 的学生加入student

5) 制作school 的差异备份 ，存入schoolDiff.bak

BACKUP DATABASE school TO DISK=’schoolDiff.bak’ WITH DIFFERENTIAL

2查看备份文件 schoolDiff.bak 中的信息

3 删除 school 数据库

4 恢复数据库 school 到第2步状态

RESTORE DATABASE school from DISK=’c:\schoolDiff.bak’WITH file=1 NORECOVERY

RESTORE DATABASE school from DISK=’c:\schoolDiff.bak’WITH file=2

Select \* from student

观察student 数据

5 恢复数据库 school 到最新状态

RESTORE DATABASE school from DISK=’c:\schoolDiff.bak’WITH file=1 NORECOVERY

RESTORE DATABASE school from DISK=’c:\schoolDiff.bak’WITH file=3

Select \* from student

观察student 数据

思考: 如果仅执行下述恢复语句,能查看 student 的数据吗?

RESTORE DATABASE school from DISK=’c:\schoolDiff.bak’WITH file=1 NORECOVERY

Select \* from student

三 利用日志备份

1 设置故障还原模型为：完全

2 建立备份

1) 制作数据文件备份 schooldata1.bak

2） 把学号 7003， 姓名：王江，性别：男，年龄为23 的学生加入student

3) 制作日志备份存入 schoollog.bak

4) 把学号 7004， 姓名：赵兰，性别：女,年龄为22 的学生加入student

5) 制作日志备份存入 schoollog.bak

3 观察schoollog中的信息

4 删除 school 数据库

5 利用schooldata1.bak 及 schoollog.bak 恢复数据库 school 到最新状态

四 使用企业管理器练习备份调度策略

1 对数据库 school 每天上午8时进行一次数据库完全备份

2 对数据库 school 的每隔 1分钟备份进行一次差异备份。

3 手工启动两个备份作业

4 删除 school 数据库

5 利用1，2步的备份进行 school的恢复。

思考：如何把备份文件备份到另外一台计算机上。

五 使用企业管理器练习数据库的分离及附加

六 如何清除日志文件。

七 使用企业管理器练习数据库的压缩

八 把school 备份到其他计算机上。

[加上 如果没有norecovert ,发生的情况]

[企业管理期恢复 如何实现 norecove ]

**试验十三 事务**

**目的:**1 掌握并理解事务

**一 理解 rollback**

1在查询分析器输入下列语句并执行 ,记录该学生的年龄。

Select \* from student where sno=’0001’

**2 执行下列 语句序列A:**

BEGIN TRANsaction

Update student set sage=sage+1 where sno=’0001’

Select \* from student where sno=’0002’

此事务结束了吗？

1. **执行：**

Select \* from student where sno=’0001’

记录该学生的年龄。

思考：student 中的0001的年龄确实被更改了吗? 为什么?

**4 执行下列语句。**

ROLLBACK TRANsaction

然后再执行：

Select \* from student where sno=’0001’

, 观察0001的年龄, 解释发生这种现象的原因。

**二 理解 commit**

1在查询分析器输入下列语句并执行 ,记录该学生的年龄。

Select \* from student where sno=’0001’

**2 执行下列 语句序列A:**

BEGIN TRANsaction

Update student set sage=sage+1 where sno=’0001’

Select \* from student where sno=’0002’

**3**执行:

commit transaction

Select \* from student where sno=’0001’

记录结果, 此时更改后的数据被永久保存了吗?

**三 执行下列 语句序列**

BEGIN TRANsaction

Update student set sage=sage+1 where sno=’0001’

Update sc set grade=grade + 1 where sno=’0002’ and cno=’1001’

Rollback

上述指令执行后，数据库发生了什么变化？

**试验十四 锁**

**目的:**1 理解锁的概念及锁的作用

一 利用帮助系统了解Sql-server 的下列语句的含义

1 锁的隔离级别

SET TRANSACTION ISOLATION LEVEL Serializable

2 设置锁定超时时间

SET LOCK\_TIMEOUT 5000

1. SP\_LOCK

**二 观察封锁**

1 执行语句序列A

BEGIN TRANsaction

Update student set sage=sage+1 where sno=’0001’

Select \* from student where sno=’0002’

2 在查询分析器中打开第二个连接(连接 school)[文件-连接], 输入下列语句:

1) select \* from student where sno=’0002’

记录执行结果,说明原因。

2）select \* from student where sno=’0001’

记录执行结果,说明原因。（如上一步没有停止，则强行终止）

3) update student set sname=’aaa’ where sno=’0002’

记录执行结果,说明原因。（如上一步没有停止，则强行终止）

4) 强行终止上一步的命令，然后执行语句：

DBCC opentran

记录结果 ,思考：如何知道此事务是那一台计算机发出的？

5）执行：

select \* from student where sno=’0001’

记录执行结果,说明原因

然后回到**第一个连接**中，执行语句:

commit Tran

观察并记录第二个连接窗口中的现象，说明原因

**三 了解锁的类型**

1 执行下列语句

BEGIN TRAN

Select \* from student where sno=’0001’

Print ‘server process ID (spid) : ‘

Print @@spid

1) 然后执行下列语句

exec sp\_lock

注意根据事务中输出的 spid ,观察结果中相应 spid 的记录, 观察加锁。

2) 然后执行下列语句

commit tran

exec sp\_lock

注意根据事务中输出的 spid ,观察结果中相应 spid 的记录, 观察加锁。

2 执行下列语句

BEGIN TRAN

Update student set sage=sage + 1 where sno=’1001’

Print ‘server process ID (spid) : ‘

Print @@spid

1) 然后执行下列语句

exec sp\_lock

注意根据事务中输出的 spid ,观察结果中相应 spid 的记录, 观察加锁。

2) 然后执行下列语句

commit tran

exec sp\_lock

注意根据事务中输出的 spid ,观察结果中相应 spid 的记录, 观察加锁。

1. 使用 SET TRANSACTION ISOLATION LEVEL Serializable

然后重新执行 三和四步，观察与原来有何不同。

1. 了解表级锁 （ 查看帮助文件）

BEGIN TRAN

Select \* from student (TABLOCKX) where sno=’1002’

Print ‘Server Process ID (spid): ‘

Print @@spid

然后执行：

exec sp\_lock

注意根据事务中输出的 spid ,观察结果中相应 spid 的记录, 观察加锁类型。

1. 了解锁定超时
   1. 执行下列语句 ，设置锁定超时为 1000 ms

set lock\_timeout 1000

go

BEGIN TRAN

Select \* from student (TABLOCKX) where sno=’1002’

2）打开第二个连接

执行：

select \* from student

记录观察到的现象。

3）在打开的第二个连接中

set lock\_timeout 10000

go

select \* from student

记录观察到的现象。

五 编程实例

1 编写存储过程 usp\_update1 , 传入参数为课程号,处理逻辑:

对传入的这门课,进行如下处理:

如某学生该门课成绩>80 , 则加 2 分

如某学生该门课成绩>60 , 则加 1 分

如某学生该门课成绩<=60 ,扣1分

要求：在存储过程中，要么全部学生的成绩被处理成功，要么全部不处理

思考：在调试中，采用那些措施，使存储过程运行时执行回滚操作 (rollback) 。

2 编写触发器，

对insert、update语句进行监控，当学生的年龄超过40岁时，把该学生的系科改为 ‘BAK’,同时删除该学生的所有选课记录。（注意，利用事务，使修改系科及删除选课记录要么全做，要么全不做）