

苏州大学实验报告

院、系	计算机学院	年级专业	19 计科图灵	姓名	张昊	学号	1927405160
课程名称	数据库课程实践					成绩	95
指导教师	赵朋朋	同组实验者	无	实验日期	2021 年 3 月 19 日		

实验名称 SQL 语言实验 1-4

试验一 数据库创建

一、实验目的

- 1 掌握利用 SQL 语言进行数据库的创建、维护。
- 2 sp_helpdb 命令

二、实验要求

- 1 创建数据库
- 2 修改数据库
- 3 删除数据库

三、实验内容

(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 。

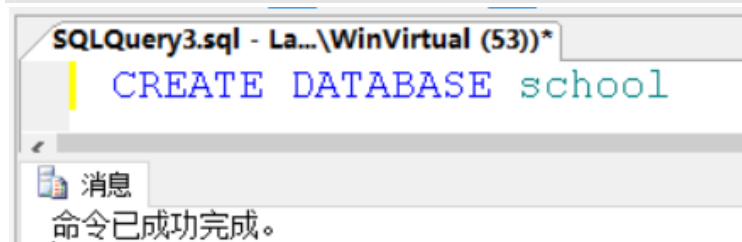
(三) create Database 深入研究

- 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%。

四、实验结果

创建数据库 school 并使用 SP_helpdb 查询数据库 school 的信息:

```
CREATE DATABASE school
```



SP_HELPDB [school]

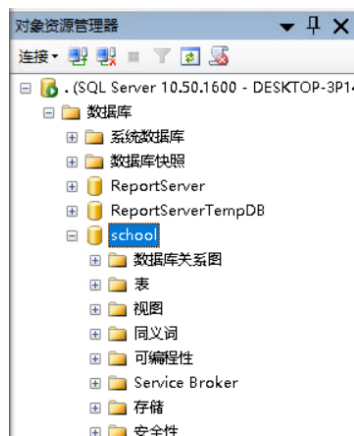
SP_HELPDB [school]

结果 消息

	name	db_size	owner	dbid	created	status	compatibility_level
1	school	2.81 MB	DESKTOP-3P14N9J\WinVirtual	7	03 19 2021	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	100

	name	fileid	filename	filegroup	size	maxsize	growth	usage
1	school	1	C:\Program Files\Microsoft SQL Server\MSSQL10_50...	PRIMARY	2304 KB	Unlimited	1024 KB	data only
2	school_log	2	C:\Program Files\Microsoft SQL Server\MSSQL10_50...	NULL	576 KB	2147483648 KB	10%	log only

使用 SQL-Server 的企业管理器查看数据库 school 的信息：



school 数据库文件所在的文件夹：

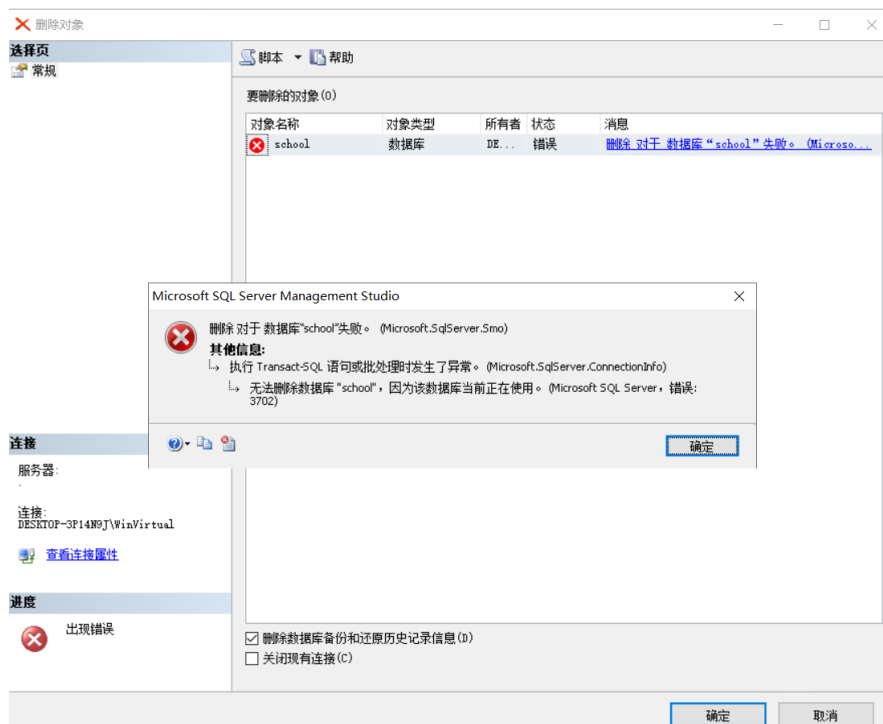
C:\Program Files\Microsoft SQL Server\MSSQL10_50.MSSQLSERVER\MSSQL\DATA

school 数据库的文件名：school.mdf

使用查询分析器删除数据库 school：

DROP DATABASE school

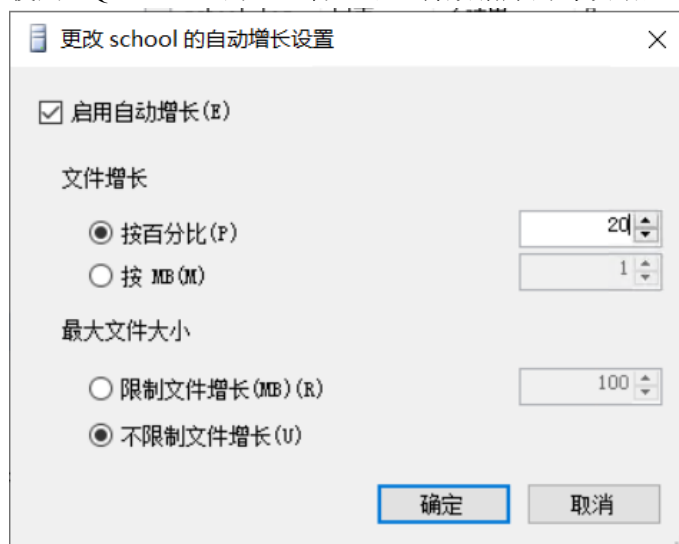
使用 SQL-Server 的企业管理器删除数据库 school：删除失败，原因：已经被删除



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:

```
CREATE DATABASE school ON (  
    NAME='school',  
    FILENAME='c:\data\school.mdf',  
    SIZE=5,  
    FILEGROWTH=1  
)
```

使用 SQL-Server 的企业管理器,将数据库的每次增量改为 20%:



五、实验总结

通过实验,我掌握了利用 SQL 语言中的 CREATE DATABASE、DROP DATABASE 命令进行数据库的创建、删除以及 sp_helpdb 命令。

试验二 创建表

一、实验目的

- 1 掌握利用 SQL 语言创建表的方法。
- 2 sp_help 命令

二、实验要求

- 1 创建表
- 2 修改表结构
- 3 删除表

三、实验内容

(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

四、实验结果

创建表：

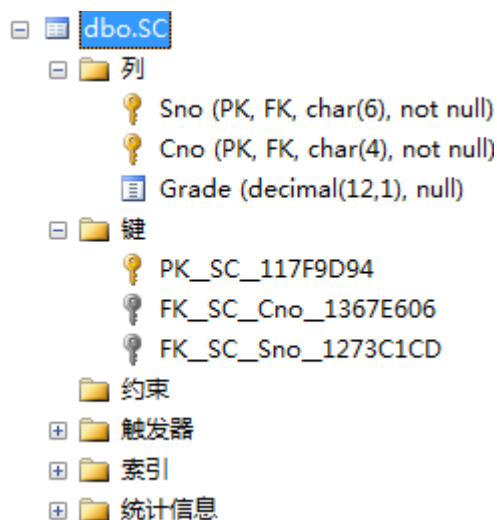
```
CREATE TABLE student (  
    Sno    char(6),  
    Sname  varchar(8),  
    Ssex   char(2),  
    Sage   smallint,  
    Sdept  varchar(15)  
)  
  
CREATE TABLE course(  
    Cno      char(4),  
    Cname    varchar(20),  
    Cpno     char(4),  
    Ccredit  tinyint  
)  
  
CREATE TABLE sc(  
    Sno      char(6),  
    Cno      char(4),  
    Grade    decimal(12,1)  
)
```

使用 SP_HELP 查看表 student 的表结构：

SP_HELP student

SP_HELP student										
结果 消息										
	Name	Owner	Type	Created_datetime						
1	student	dbo	user table	2021-03-19 20:17:21.740						
	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	Sno	char	no	6			yes	no	yes	Chinese_PRC_CI_AS
2	Sname	varchar	no	8			yes	no	yes	Chinese_PRC_CI_AS
3	Ssex	char	no	2			yes	no	yes	Chinese_PRC_CI_AS
4	Sage	smallint	no	2	5	0	yes	(n/a)	(n/a)	NULL
5	Sdept	varchar	no	15			yes	no	yes	Chinese_PRC_CI_AS
	Identity	Seed	Increment	Not For Replication						
1	No identity column defined.	NULL	NULL	NULL						
	RowGuidCol									
1	No rowguidcol column defined.									
	Data_located_on_filegroup									
1	PRIMARY									

利用企业管理器查看表 sc 的表结构：



利用 sql 语句表结构修改

在 student 表中添加列：

```
ALTER TABLE student ADD address varchar(60)
```

```
ALTER TABLE student ADD inDate datetime
```

SP_HELP student

	Name	Owner	Type	Created_datetime
1	student	dbo	user table	2021-03-19 20:51:21.010

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	Sno	char	no	6			yes	no	yes	Chinese_PRC_CI_AS
2	Sname	varchar	no	8			yes	no	yes	Chinese_PRC_CI_AS
3	Ssex	char	no	2			yes	no	yes	Chinese_PRC_CI_AS
4	Sage	smallint	no	2	5	0	yes	(n/a)	(n/a)	NULL
5	Sdept	varchar	no	15			yes	no	yes	Chinese_PRC_CI_AS
6	address	varchar	no	60			yes	no	yes	Chinese_PRC_CI_AS
7	inDate	datetime	no	8			yes	(n/a)	(n/a)	NULL

将家庭地址 address 长度为 50:

```
ALTER TABLE student ALTER COLUMN address varchar(50)
```

SP_HELP student

	Name	Owner	Type	Created_datetime
1	student	dbo	user table	2021-03-19 20:51:21.010

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	Sno	char	no	6			yes	no	yes	Chinese_PRC_CI_AS
2	Sname	varchar	no	8			yes	no	yes	Chinese_PRC_CI_AS
3	Ssex	char	no	2			yes	no	yes	Chinese_PRC_CI_AS
4	Sage	smallint	no	2	5	0	yes	(n/a)	(n/a)	NULL
5	Sdept	varchar	no	15			yes	no	yes	Chinese_PRC_CI_AS
6	address	varchar	no	50			yes	no	yes	Chinese_PRC_CI_AS
7	inDate	datetime	no	8			yes	(n/a)	(n/a)	NULL

删除 student 表的 inDate 列：

```
ALTER TABLE student DROP COLUMN inDate
```

SP_HELP student

Name	Owner	Type	Created_datetime
1 student	dbo	user table	2021-03-19 20:51:21.010

Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1 Sno	char	no	6			yes	no	yes	Chinese_PRC_CI_AS
2 Sname	varchar	no	8			yes	no	yes	Chinese_PRC_CI_AS
3 Ssex	char	no	2			yes	no	yes	Chinese_PRC_CI_AS
4 Sage	smallint	no	2	5	0	yes	(n/a)	(n/a)	NULL
5 Sdept	varchar	no	15			yes	no	yes	Chinese_PRC_CI_AS
6 address	varchar	no	50			yes	no	yes	Chinese_PRC_CI_AS

删除表：

```
DROP TABLE sc
DROP TABLE student
DROP TABLE course
```

五、实验总结

通过实验，我掌握了 SQL 语言中创建表的 CREATE TABLE 语句，在已有的表中添加、修改或删除列的 ALTER TABLE 语句和删除表的 DROP TABLE 语句；熟悉了 sp_help 命令。

试验三 创建数据完整性

一、实验目的

- 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。

二、实验要求

- 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

三、实验内容

(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

- 2 **Course** 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。

- 4 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

四、实验结果

带有完整性约束的 Create Table 命令建立表 student、course、sc:

```
CREATE TABLE student (
    Sno    char(6)          PRIMARY KEY,
    Sname  varchar(8)       NOT NULL UNIQUE,
    Ssex   char(2)          NOT NULL CHECK (Ssex = '男' OR Ssex = '女'),
    Sage   smallint         CHECK (Sage > 16),
    Sdept  varchar(15)      NOT NULL DEFAULT 'JSJ'
)

CREATE TABLE course(
    Cno     char(4)          PRIMARY KEY,
    Cname   varchar(20),
    Cpno    char(4),
    Ccredit tinyint         CHECK (Ccredit >= 0 AND Ccredit <= 5),

    CHECK (Cno <> Cpno)
)

CREATE TABLE sc(
    Sno     char(6),
    Cno     char(4),
    Grade   decimal(12,1),

    CONSTRAINT PK_SC PRIMARY KEY (Sno, Cno),
    FOREIGN KEY (Sno) REFERENCES student(Sno),
    FOREIGN KEY (Cno) REFERENCES course(Cno)
)
```

SP_HELP [student]

Name	Owner	Type	Created_datetime
1 student	dbo	user table	2021-03-20 15:34:20.550

Column_name	Type	Computed	Length	Prec	Scale	Nullable	Trim TrailingBlanks	FixedLenNullInSource	Collation
1 Sno	char	no	6			no	no	no	Chinese_PRC_CI_AS
2 Sname	varchar	no	8			no	no	no	Chinese_PRC_CI_AS
3 Ssex	char	no	2			no	no	no	Chinese_PRC_CI_AS
4 Sage	smallint	no	2	5	0	yes	(n/a)	(n/a)	NULL
5 Sdept	varchar	no	15			no	no	no	Chinese_PRC_CI_AS

Identity	Seed	Increment	Not For Replication
1 No identity column defined.	NULL	NULL	NULL

RowGuidCol
1 No rowguidcol column defined.

Data_located_on_filegroup
1 PRIMARY

index_name	index_description	index_keys
1 PK_student__CA1FE4647F60ED59	clustered, unique, primary key located on PRIMARY	Sno
2 UQ_student__52723D27023D5A04	nonclustered, unique, unique key located on PRIMA...	Sname

constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
1 CHECK on column Sage	CK_student__Sage__0519C6AF	(n/a)	(n/a)	Enabled	Is_For_Replication	[(Sage)>(16)]
2 CHECK on column Ssex	CK_student__Ssex__0425A276	(n/a)	(n/a)	Enabled	Is_For_Replication	[(Ssex]='男' OR [Ssex]='女')
3 DEFAULT on column Sdept	DF_student__Sdept__060DEAE8	(n/a)	(n/a)	(n/a)	(n/a)	(JSJ)
4 PRIMARY KEY (clustered)	PK_student__CA1FE4647F60ED59	(n/a)	(n/a)	(n/a)	(n/a)	Sno
5 UNIQUE (non-clustered)	UQ_student__52723D27023D5A04	(n/a)	(n/a)	(n/a)	(n/a)	Sname

Table is referenced by foreign key
1 school.dbo.sc: FK_sc__Sno__0EA330E9

Table is referenced by views

SP_HELP [sc]

Name	Owner	Type	Created_datetime
1 sc	dbo	user table	2021-03-20 15:34:20.560

Column_name	Type	Computed	Length	Prec	Scale	Nullable	Trim TrailingBlanks	FixedLenNullInSource	Collation
1 Sno	char	no	6			no	no	no	Chinese_PRC_CI_AS
2 Cno	char	no	4			no	no	no	Chinese_PRC_CI_AS
3 Grade	decimal	no	9	12	1	yes	(n/a)	(n/a)	NULL

Identity	Seed	Increment	Not For Replication
1 No identity column defined.	NULL	NULL	NULL

RowGuidCol
1 No rowguidcol column defined.

Data_located_on_filegroup
1 PRIMARY

index_name	index_description	index_keys
1 PK_SC	clustered, unique, primary key located on PRIMARY	Sno, Cno

constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
1 FOREIGN KEY	FK_sc__Cno__0F975522	No Action	No Action	Enabled	Is_For_Replication	Cno
2						REFERENCES school.dbo.course (Cno)
3 FOREIGN KEY	FK_sc__Sno__0EA330E9	No Action	No Action	Enabled	Is_For_Replication	Sno
4						REFERENCES school.dbo.student (Sno)
5 PRIMARY K...	PK_SC	(n/a)	(n/a)	(n/a)	(n/a)	Sno, Cno

删除 sc 的主码:

ALTER TABLE sc DROP PK_SC

删除 SC 表参照 course 表的外码:

① 先用 SP_HELP 找到参照 course 表的外码名, 为 FK_sc__Cno__0F975522

SP_HELP [sc]

结果 消息

Name	Owner	Type	Created_datetime
1 sc	dbo	user table	2021-03-20 15:34:20.560

Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1 Sno	char	no	6			no	no	no	Chinese_PRC_CI_AS
2 Cno	char	no	4			no	no	no	Chinese_PRC_CI_AS
3 Grade	decimal	no	9	12	1	yes	(n/a)	(n/a)	NULL

Identity	Seed	Increment	Not For Replication
1 No identity column defined.	NULL	NULL	NULL

RowGuidCol
1 No rowguidcol column defined.

Data_located_on_filegroup
1 PRIMARY

constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
1 FOREIGN KEY	FK_sc_Cno_0F975522	No Action	No Action	Enabled	Is_For_Replication	Cno
2						REFERENCES school.dbo.course (Cno)
3 FOREIGN KEY	FK_sc_Sno_0EA330E9	No Action	No Action	Enabled	Is_For_Replication	Sno
4						REFERENCES school.dbo.student (Sno)

② 再用 ALTER TABLE 删除外码

```
ALTER TABLE sc DROP FK_sc_Cno_0F975522
```

添加 sc 表的主码:

```
ALTER TABLE sc ADD CONSTRAINT PK_SC PRIMARY KEY (Sno, Cno)
```

添加 sc 表的 Cno 的外码, 参照表 course 的 Cno:

```
ALTER TABLE sc ADD FOREIGN KEY (Cno) REFERENCES course(Cno)
```

加自定义约束: 表 sc 的成绩只能在 0-100 分之间:

```
ALTER TABLE sc ADD CHECK (Grade >= 0 AND Grade <= 100)
```

SP_HELP [sc]

结果 消息

Name	Owner	Type	Created_datetime
1 sc	dbo	user table	2021-03-20 15:34:20.560

Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1 Sno	char	no	6			no	no	no	Chinese_PRC_CI_AS
2 Cno	char	no	4			no	no	no	Chinese_PRC_CI_AS
3 Grade	decimal	no	9	12	1	yes	(n/a)	(n/a)	NULL

Identity	Seed	Increment	Not For Replication
1 No identity column defined.	NULL	NULL	NULL

RowGuidCol
1 No rowguidcol column defined.

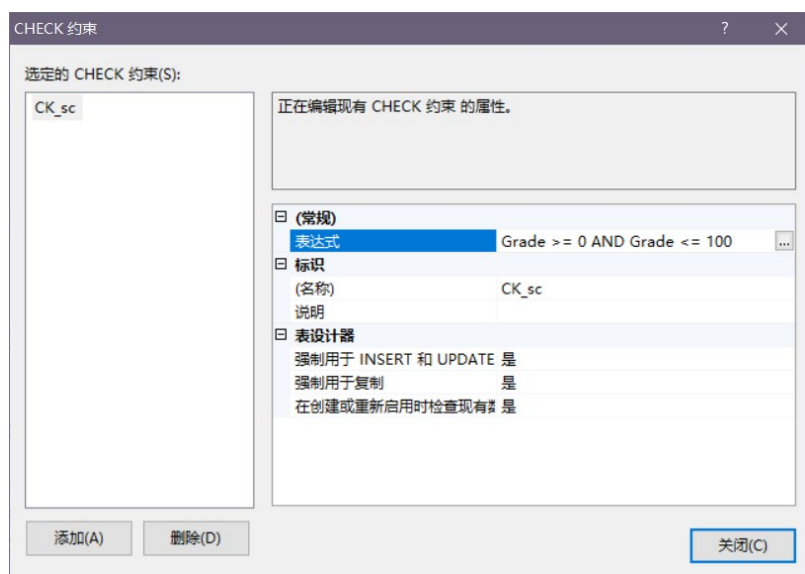
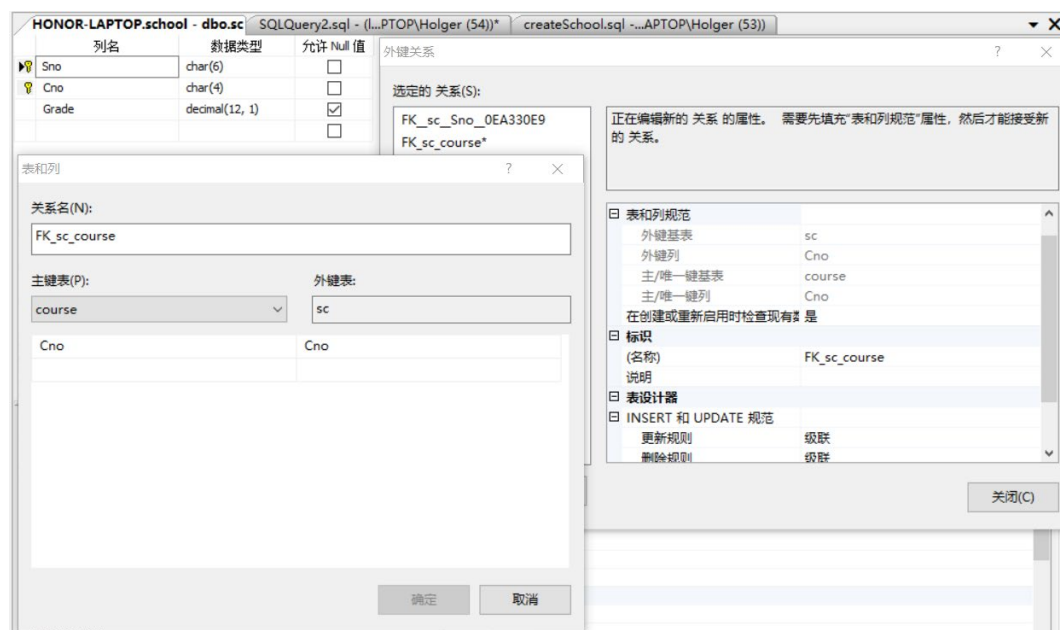
Data_located_on_filegroup
1 PRIMARY

index_name	index_description	index_keys
1 PK_SC	clustered, unique, primary key located on PRIMARY	Sno, Cno

constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
1 CHECK on column Grade	CK_sc_Grade_1273C1CD	(n/a)	(n/a)	Enabled	Is_For_Replication	([Grade]>=(0) AND [Grade]<=(100))
2 FOREIGN KEY	FK_sc_Cno_117F9D94	No Action	No Action	Enabled	Is_For_Replication	Cno
3						REFERENCES school.dbo.course (Cno)
4 FOREIGN KEY	FK_sc_Sno_0EA330E9	No Action	No Action	Enabled	Is_For_Replication	Sno
5						REFERENCES school.dbo.student (Sno)
6 PRIMARY KEY (clustered)	PK_SC	(n/a)	(n/a)	(n/a)	(n/a)	Sno, Cno

Table is referenced by views

使用企业管理器更改：



查看信息:

```
select * from student
select * from course
select * from sc
```

结果		消息		
Sno	Sname	Ssex	Sage	Sdept
Cno	Cname	Cpno	Ccredit	
Sno	Cno	Grade		

五、实验总结

通过实验我掌握了创建数据完整性约束的命令，了解到修改、删除表的完整性约束的方法。

试验四 数据完整性试验

一、实验目的

- 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。

二、实验要求

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`。

三、实验内容

(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 course` 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 sc` 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。

四、实验结果

(一) 实体完整性

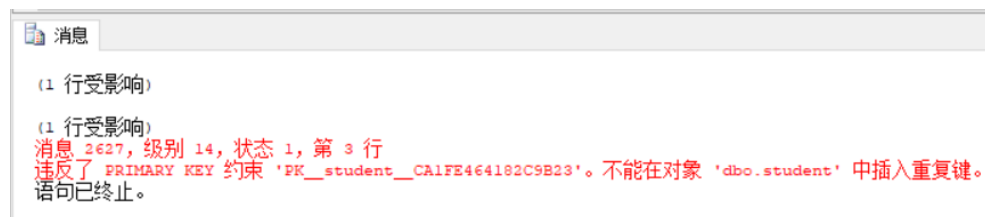
1 student 表数据输入:

```
INSERT INTO student VALUES (3001, '赵达', '男', 20, 'SX')
INSERT INTO student VALUES (3002, '杨丽', '女', 21, 'JSJ')
INSERT INTO student VALUES (3001, '李寅', '女', 21, 'SX')
```

数据“李寅”

3001	李寅	女	21	SX
------	----	---	----	----

插入失败, 与“赵达”主键重复:



输入了两条数据:

```
SELECT * FROM student
```


	Sno	Sname	Ssex	Sage	Sdept
1	3001	赵达	男	20	SX
2	3002	杨丽	女	21	JSJ

2 course 表数据的输入:

```
INSERT INTO course VALUES (1081, '电子商务', NULL, 4)
```

输入成功:

消息
(1 行受影响)

3 SC 表数据的输入:

```
INSERT INTO sc VALUES (3001, 1081, 90)
```

```
INSERT INTO sc VALUES (3001, 1081, 79)
```

第二条数据

3001	1081	79
------	------	----

输入失败, 主键重复:

消息
(1 行受影响)
消息 2627, 级别 14, 状态 1, 第 2 行 违反了 PRIMARY KEY 约束 'PK_SC'。不能在对象 'dbo.sc' 中插入重复键。 语句已终止。

(二) 用户自定义完整性约束

1 student 表数据输入:


```
INSERT INTO student VALUES (3005, '赵达', '男', 14, 'SX')
```

```
INSERT INTO student VALUES (3006, '杨丽', '南', 21, 'JSJ')
```

插入均失败, “赵达”年龄不符合约束 Sage>16, “杨丽”性别不为 {男, 女} 中的一个。

消息
消息 547, 级别 16, 状态 0, 第 1 行 INSERT 语句与 CHECK 约束“CK__student__Sage__32E0915F”冲突。该冲突发生于数据库“school”, 表“dbo.student”, column 'Sage'。 语句已终止。
消息 547, 级别 16, 状态 0, 第 2 行 INSERT 语句与 CHECK 约束“CK__student__Ssex__31EC6D26”冲突。该冲突发生于数据库“school”, 表“dbo.student”, column 'Ssex'。 语句已终止。

```
select * from student
```

	结果		消息		
	Sno	Sname	Ssex	Sage	Sdept

没有成功输入数据, student 表为空

2 course 表数据的输入:

```
INSERT INTO course VALUES (1085, 'C++', NULL, 9)
```



```
INSERT INTO course VALUES (1086, '语文', 1086, 3)
```

插入均失败, C++学分太大, 不满足约束 Ccredit 取值范围 {0,1,2,3,4,5}; “语文”课程的先

修课程不能为自身，不满足约束每一行的 Cno 与 cpno 不可相同：

消息
消息 547, 级别 16, 状态 0, 第 1 行
INSERT 语句与 CHECK 约束"CK_course_Ccredit_38996AB5"冲突。该冲突发生于数据库"school", 表"dbo.course", column 'Ccredit'。
语句已终止。
消息 547, 级别 16, 状态 0, 第 2 行
INSERT 语句与 CHECK 约束"CK_course_398D9EEE"冲突。该冲突发生于数据库"school", 表"dbo.course"。
语句已终止。

```
select * from course
```

 结果	 消息		
Cno	Cname	Cpno	Ccredit

没有成功输入数据，course 表为空


3 SC 表数据的输入：

```
INSERT INTO sc VALUES (3002, 1081, 128)
```

插入失败，sc 表的 Sno 是外码，参照表 student 的 Sno，但 student 表没有学号为 3002 这个学生（因为 student 表为空）：

消息
消息 547, 级别 16, 状态 0, 第 1 行
INSERT 语句与 FOREIGN KEY 约束"FK_sc_Sno_3C69FB99"冲突。该冲突发生于数据库"school", 表"dbo.student", column 'Sno'。
语句已终止。

```
select * from sc
```

 结果		 消息	
Sno	Cno	Grade	

没有成功插入数据，sc 表为空

（三）参照完整性约束

输入实验前的数据：

```
INSERT INTO student VALUES (4001, '赵尹', '男', 20, 'SX')
INSERT INTO student VALUES (4002, '杨开', '女', 20, 'JSJ')
INSERT INTO course VALUES (1088, 'Java', NULL, 5)
INSERT INTO course VALUES (1089, '数学', NULL, 3)
INSERT INTO sc VALUES (4001, 1088, 90)
INSERT INTO sc VALUES (4002, 1088, 86)
```

消息
(1 行受影响)
(1 行受影响)
(1 行受影响)
(1 行受影响)
(1 行受影响)
(1 行受影响)

1) 在 SC 表中添加新行：

```
INSERT INTO sc VALUES (4001, 1066, 76)
```

插入失败，sc 表的 Cno 是外码，参照表 course 的 Cno，但 course 表没有编号为 1066 的课程：



消息 547, 级别 16, 状态 0, 第 1 行
INSERT 语句与 FOREIGN KEY 约束“FK__sc__Cno__3D5E1FD2”冲突。该冲突发生于数据库“school”, 表“dbo.course”, column ‘Cno’。
语句已终止。

2) 在 student 表中添加新行:

```
INSERT INTO student VALUES (4003, '赵辉', '男', 21, 'SX')
```

成功:



(1 行受影响)

3) 删除 student 表的 4001, 4002 学生:

```
DELETE FROM student WHERE Sno = 4001
```

```
DELETE FROM student WHERE Sno = 4002
```

不能删除, 限制在于 sc 表中有参照 student 表的外键 Sno, 换句话说, sc 表中还有 4001, 4002 学生的选课记录:



消息 547, 级别 16, 状态 0, 第 1 行
DELETE 语句与 REFERENCE 约束“FK__sc__Sno__3C69FB99”冲突。该冲突发生于数据库“school”, 表“dbo.sc”, column ‘Sno’。
语句已终止。
消息 547, 级别 16, 状态 0, 第 2 行
DELETE 语句与 REFERENCE 约束“FK__sc__Sno__3C69FB99”冲突。该冲突发生于数据库“school”, 表“dbo.sc”, column ‘Sno’。
语句已终止。

将 SC 的参照 student 的外键改成级联删除即可执行:

①先使用 SP_HELP 找到 sc 参照 student 的外键 Sno: FK__sc__Sno__3C69FB99

②删除该外键并建立级联删除的外键

```
ALTER TABLE sc DROP FK__sc__Sno__3C69FB99
```

```
ALTER TABLE sc ADD FOREIGN KEY (Sno)
```

```
REFERENCES student(Sno) ON DELETE CASCADE
```

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

```
UPDATE student SET Sno=4018 WHERE Sno=4003
```

```
UPDATE student SET Sno=4021 WHERE Sno=4001
```

学号 4003 成功修改, 因为 sc 表中没有学号为 4003 的记录

学号 4001 修改失败, 因为 sc 表中有学号为 4001 的记录

原因在于 sc 表中有参照 student 表的外键 Sno



(1 行受影响)
消息 547, 级别 16, 状态 0, 第 2 行
UPDATE 语句与 REFERENCE 约束“FK__sc__Sno__3E52440B”冲突。该冲突发生于数据库“school”, 表“dbo.sc”, column ‘Sno’。
语句已终止。

5) 将 SC 的参照 student 的外键改成级联修改即可执行:

①先使用 SP_HELP 找到 sc 参照 student 的外键 Sno: FK__sc__Sno__3E52440B

②删除该外键并建立级联修改的外键

```
ALTER TABLE sc DROP FK__sc__Sno__3E52440B
```

```
ALTER TABLE sc ADD FOREIGN KEY (Sno)
```

```
REFERENCES student(Sno) ON UPDATE CASCADE
```

sc 表中学号从 4001 改为 4011:

```
UPDATE sc SET Sno=4011 WHERE Sno=4001 AND Cno=1088 AND Grade=90
```

修改不成功, sc 的外码 Sno 参照 student 表的 Sno, 但学生 4011 不存在 student 表中:



消息

消息 547, 级别 16, 状态 0, 第 1 行

UPDATE 语句与 FOREIGN KEY 约束"FK__sc__Sno__3F466844"冲突。该冲突发生于数据库"school", 表"dbo.student", column 'Sno'。

语句已终止。

可以采取在 student 表中新增学号为 4011 的学生记录

将 4001 修改为 4003 能成功

在 sc 表中, 外码不可以为空, 因为 sc 表的主码是 {Sno, Cno}, 需要参照 student 的 Sno 和 course 的 Cno 才能保证数据的完整和一致, 故需要外码的存在。外码也可以为空, 例如 student 表和 course 表。

五、实验总结

通过实验, 我理解了实体完整性、参照完整性、用户自定义完整性的作用, 掌握了外码的作用, 掌握了级联修改、级联删除的概念。