表操作

向表中添加列

alter table couse add 选课日期 datetime null

删除表中指定的列

ALTER TABLE dept DROP COLUMN dp\_count

数据完整性

### 空值约束( NULL )

### 操作3.1：将student表中的st\_sex列属性更改为NOT NULL

### 参考答案：

### ALTER TABLE student ALTER COLUMN st\_sex nVarChar(8) NOT NULL

### 空值约束( NULL )

操作3.1：将student表中的st\_sex列属性更改为NOT NULL

参考答案：

ALTER TABLE student ALTER COLUMN st\_sex nVarChar(8) NOT NULL

### 3．默认值对象空值约束( NULL )

### 操作3.1：将student表中的st\_sex列属性更改为NOT NULL

### 参考答案：

### ALTER TABLE student ALTER COLUMN st\_sex nVarChar(8) NOT NULL

操作3.3：创建默认值对象df\_today为当前日期，并将其绑定到couse表中的‘选课日期’列，然后取消绑定，最后删除默认值对象df\_today。

create default df\_today as getdate() //创建默认对象

go

exec sp\_bindefault df\_today,'couse.选课日期' //绑定到选课日期列

go

exec sp\_unbindefault 'couse.选课日期' //取消绑定

go

drop default df\_today //删除默认对象

Go

1. 检查约束 将couse 表中的 分数列 约束值为>=0或<=100

alter table couse add check(分数>=0 and 分数<=100)

5.规则约束对象，将student 表中的性别只允许设置为 男 或 女

创建规则约束对象rl\_sex，用于检查性别的取值仅限于“男”和“女”，并将其绑定到student表中的st\_sex列，然后取消绑定，最后删除规则约束对象rl\_sex

create rule r3\_sex as @学生性别='男' or @学生性别= '女'

go

EXEC sp\_bindrule r3\_sex, 'student.学生性别'

GO

EXEC sp\_unbindrule 'student.学生性别'

GO

DROP RULE r3\_sex

GO

Student 表中的性别只允许男或女

alter table student

add constraint r2\_sex check(学生性别 in('男','女'))

**二、创建数据库**

用T-SQL创建一个名为stj的数据库，一个数据文件一个日志文件，数据文件逻辑名为stj\_data，物理文件名分别为stj\_data.mdf，初始大小为5MB；日志文件逻辑名为stj\_log，物理文件名为stj\_log.ldf，初始大小为1MB，最大大小为7MB，增长速度为2MB。

create database stj

on

(

name="stj\_data",

filename="E:\sql2012\stj\_data.dmf",

size=5mb,

filegrowth=1mb

)

log on

(

name="stj\_log",

filename="E:\sql2012\stj\_lod.ldf",

size=1mb,

maxsize=7mb,

filegrowth=2mb

)

use stj

Go

**三、表操作**

stj数据库包括三个关系：

学生(学号,姓名,性别, 年龄, 系别)

课程表(课程号,课程名)

选课表（学号，课程号，成绩）

1、用SQL语句定义上述表的结构

2、为三个表添加主键约束

3、为学生表姓名列添加唯一性约束

4、为选课表添加check约束,成绩输入只能在0-100之间

4、插入记录

create table 学生表

(

学号 char(10) not null,

姓名 char(10) not null,

性别 bit not null,

年龄 char(10) null,

系别 char(10) null

)

create table 课程表

(

课程号 char(10) not null,

课程名 char(10) not null

)

create table 选课表

(

学号 char(10) not null,

课程号 char(10) null,

成绩 char(10) null

)

添加主键约束

alter table 学生表 add constraint id primary key(姓名)

alter table 课程表 add constraint pd primary key(课程号)

alter table 选课表 add constraint ld primary key(学号)

Check 约束

alter table 选课表 add check(成绩>=0 and 成绩<=100)

唯一性约束

alter table 学生表 add unique(姓名)

插入数据

insert into 课程表

values('1','SQL Serves','4'),

('2','VB','3'),

('3','专业英语','2')

go

insert into 学生表

values('1','李明',0,'20','计算机系'),

('2','张然',1,'19','经管系'),

('3','许晴',1,'18','电子系')

go

insert into 选课表

values('1','1','90'),

('2','1','85'),

('3','2','76'),

('4','3','70')

Go

1. 把课程表中课程号为2的课程的学分修改为4。

update 课程表

set 学分='4'

where 课程号='2'

go

**四、查询**

1. 查询选修表中选修1课程的学生学号

select 学号 as number

from 选课表

where 课程号='1'

1. 列出所有姓“李”且单名的学生

select \*

from 学生表

where 姓名 like '李\_'

1. 统计男女学生的人数

select 性别,count(\*) as'人数'

from 学生表

group by rollup(性别)

1. 查询学生选修课程成绩大于80分的学生情况。

select 学号,成绩

from 选课表

where 成绩>=80

1. 查询选修了课程名为VB”的学生学号和姓名。

select 学生表.学号,姓名

from 学生表,选课表,课程表

where 学生表.学号=选课表.学号

and 选课表.课程号=课程表.课程号

and 课程表.课程名='VB'

1. 创建视图，检索“选修了SQL Server”课程的学生信息以及成绩。

create view ccj1 with encryption

as

select 学生表.学号,姓名,性别,年龄,选课表.成绩

from 学生表,选课表,课程表

where 学生表.学号=选课表.学号 and 课程名='SQL Serves'

with check option

///计算机系同学的学号，姓名和总学分

Select 学号 as number,姓名 as name，总学分as mark

From xsb

Where 专业=’计算机’

///查找总分大于80分的人数

select count(\*) as '总分>80分的人数'

from 选课表

where 成绩>80;

go

1. 创建一存储过程getDetailByName，通过输入参数学生姓名（如“张然"），筛选出该学生的基本信息,对不存在此学生姓名的输入值，必须作一检测，打印信息“不存在此学生”。

Create procedure getDetailByName @Name nvarchar(10)

AS

IF (SELECT COUNT(\*) FROM 学生表 WHERE 姓名 = @Name) > 0

SELECT \*

FROM 学生表 Where 姓名 = @Name

ELSE

print '警示 ： 不存在姓名为 ' + @Name + ' 的学生资料'

execute getDetailByName '张然'

1. 创建触发器（INSERT,UPDATE触发器也要看）

在学生表中创建DELETE触发器，当删除学生表中某一学号时，同时删除选课表中该学生的相关信息。

create trigger xsb

on 学生表 after delete

as

begin

delete from 学生表

where 学号 in(select 学号 from deleted)

End

delete from 学生表

where 学号='1'

Go

3.创建insert 触发器，当向表中插入一名学生的成绩时，将表中该学生的总学分加上添加的课程的学分

Create trigger cjb\_insert

On cjb after insert

As

Beggin

Declare @num char(6),@kc\_num char(3)

Declare @xf int

Select @num=学号,@kc\_num=课程号 from inserted

Select @xf=学分 from kcb where 课程号=@kc\_num

Update xsb set 总学分=总学分+@xf where 学号+@name

Print’修改成功’

end