--1.建数据库dbspj,并保存.

create database dbspj

on primary

(name='dbspj',

filename='C:\dbspj.mdf',

size=10mb,

filegrowth=10mb)

log on

(name='dbspj.log',

filename='C:\dbspj\_log.ldf',

size=5mb,

filegrowth=10%)

use dbspj

---------------------------------------实体完整性-------------

----1.建表供货商s,属性有sno,sname,city.

drop table s

CREATE TABLE s

(SNO nvarchar(9) primary key not null,

SNAME nvarchar(20) not null,

CITY nvarchar(20)

);

--2.插入如下数据.请举例: 插入一个新的数据,但违反实体完整性,不可插入的例子

insert into s values('S1','精益','天津');

insert into s values('S2','盛锡','北京');

insert into s values('S3','东方红','北京');

insert into s values('S4','丰泰盛','天津');

insert into s values('S5','为民','天津');

--insert into s values('','为民','天津');----违反实体完整性???为何可以插入?? not null也不管用

--insert into s values('S5','','天津1');--主码不能重复

select \* from s

----3.建零件表p, 属性有pno,pname ,color,weight.

drop table p

CREATE TABLE P

(PNO nvarchar(9) primary key check([pno]is not null and [pno]<>''),

PNAME nvarchar(20) ,

COLOR nvarchar(20),

WEIGHT SMALLINT

);

----4,插入下列数据后,请举例: 违反实体完整性而不能插入的例子

insert into P values('P1','螺母','红',12);

insert into P values('P2','螺栓','绿',17);

insert into P values('P3','螺丝刀','蓝',14);

insert into P values('P4','螺丝刀','红',14);

insert into P values('P5','凸轮','蓝',40);

insert into P values('P6','齿轮','红',30);

insert into P values('','齿轮','红',30);-----------PNO nvarchar(9) primary key not null时,能执行

insert into P values('','齿轮','红',30);-----------PNO nvarchar(9) primary key check([pno]is not null and [pno]<>''),不能执行

insert into P values('P6','齿轮','红',30);-- 不可以执行,主码值不唯一

select \* from P

-----5 ,建客户表J，属性有jno ,jname,city,jname要求值唯一,不能重复

drop table J

CREATE TABLE J

(JNO nvarchar(9) PRIMARY KEY not null,

JNAME nvarchar(20) UNIQUE,

CITY nvarchar(20)

);

insert into J values('J1','三建','北京');

insert into J values('J2','一汽','长春');

insert into J values('J3','弹簧厂','天津');

insert into J values('J4','造船厂','天津');

insert into J values('J5','机车厂','唐山');

insert into J values('J6','无线电厂','常州');

insert into J values('J7','半导体厂','南京');

insert into J values('J6','半导体厂','南京');

insert into J values('J7','半导体厂','南京');-------错误-------违反了 PRIMARY KEY 约束“PK\_\_J\_\_C4DECBEE9068AD60”。不能在对象“dbo.J”中插入重复键。重复键值为 (J7)。

insert into J values('','1','2')

select \* from J

------------------------------------------参照完整性----------------------------------

------6. 建采购单表spj,属性有sno ,pno,jno,qty(数量）

drop table spj

CREATE TABLE SPJ

( SNO nvarchar(9),

PNO nvarchar(9),

JNO nvarchar(9),

QTY SMALLINT,

primary key(sno,pno,jno),---主码不是一个属性,只能在表级定义主码

foreign key (sno) references s(sno),----3个外键

foreign key (pno) references p(pno),

foreign key (jno) references j(jno)

);

----7.插入下列数据后,请举例1: 违反参照完整性而无法插入新数据的例子.举例2: 违反实体完整性而无法插入新数据的例子

insert into SPJ values('S1','P1','J1',200);

insert into SPJ values('S1','P1','J3',100);

insert into SPJ values('S1','P1','J4',700);

insert into SPJ values('S1','P2','J2',100);

insert into SPJ values('S2','P3','J1',400);

insert into SPJ values('S2','P3','J2',200);

insert into SPJ values('S2','P3','J4',500);

insert into SPJ values('S2','P3','J5',400);

insert into SPJ values('S2','P5','J1',400);

insert into SPJ values('S2','P5','J2',100);

insert into SPJ values('S2','P5','J2',100);

insert into SPJ values('S3','P1','J1',200);

insert into SPJ values('S3','P3','J1',200);

insert into SPJ values('S4','P5','J1',100);

insert into SPJ values('S4','P6','J3',300);

insert into SPJ values('S4','P6','J4',200);

insert into SPJ values('S5','P2','J4',100);

insert into SPJ values('S5','P3','J1',200);

insert into SPJ values('S5','P6','J2',200);

insert into SPJ values('S5','P6','J4',500);

---insert into SPJ values('S10','P1','J1',200);----违反了参照完整性,无S10

--insert into SPJ values('S5','P6','J4',200);-------违反了实体完整性

select \* from SPJ

---8.运行drop table s ,这时候可以删除s吗?

drop table s -----无法删除对象 's'，因为该对象正由一个 FOREIGN KEY 约束引用。

drop table spj

drop table s---可以删除s了

-------------------------------用户定义的完整性---------------------------------

--9. 建立表SPJ,属性有sno ,pno,jno,qty(数量）,要求qty在0到100之间

drop table spj

CREATE TABLE SPJ

( SNO nvarchar(9),

PNO nvarchar(9),

JNO nvarchar(9),

QTY SMALLINT check (qty>=0 and qty<=100),

primary key(sno,pno,jno),---主码不是一个属性,只能在表级定义主码

foreign key (sno) references s(sno),----3个外键

foreign key (pno) references p(pno),

foreign key (jno) references j(jno)

);

---10.举例：违反用户定义的完整性导致不能插入的例子

insert into SPJ values('S5','P6','J4',700);-----与check约束冲突

--------------------------------完整性约束命名子句-----------------------------

--1.建立教师信息表，要求每个教师应发工资>=3000，应发工资是工资列和扣除项deduct之和。教师号在1000-9999之间，姓名不能取空值，性别只能是男or女

drop table teacher

create table teacher

(eno numeric(4) primary key constraint c1 check(eno between 1000 and 9997),

ename char (10) constraint c2 not null,

sal numeric(7,2),

deduct numeric(7,2),

esex char(2) constraint c3 check (esex in ('男','女')),

constraint c4 check (sal+deduct >=3000)

);

insert into teacher values(9999,'d',2000,3000,'男')---INSERT 语句与 CHECK 约束"c1"冲突,不可以增加。

select \* from teacher

--2.删除教师信息表中的教师号在1000-9999的限制

alter table teacher

drop constraint c1

insert into teacher values(9999,'d',2000,3000,'男')---可以增加

----------------------------------------------触发器准备代码-------------------

create database dbstu1

on primary

( name='dbstu1',

filename='C:\dbstu1.mdf',

size=10mb,

filegrowth=10mb)

log on

(name='dbstu1.log',

filename='C:\dbstu1\_log.ldf',

size=5mb,

filegrowth=10%

)

----2.用脚本创建dbstu1中的表 classinfo（编号，名称），--要求编号从0开始自动加1-----建表

use dbstu1

create table classinfo

( cid int not null primary key identity(0,1),

ctitle nvarchar(10)

)

select \* from classinfo

---3.向classinfo表中批量插入5条数据------添加数据

---方式1，不指定列名

insert into classinfo

values ('计科181'),('计科182'),('网络181'),('网络182'),('计科183')

select \* from classinfo

----方式2，指定列名

insert into classinfo(ctitle)

values ('计科181'),('计科182'),('网络181'),('网络182'),('计科183')

select \* from classinfo

-----4.用脚本创建dbstu1中的表 studentinfo（编号，姓名，性别，出生日期，电话，邮箱，班级）

use dbstu1

create table studentinfo

(sid int not null primary key ,

sname nvarchar(10),

sgender bit ,

sbirth date,

sphone char (11),

semail varchar (20),

cid int not null foreign key (cid) references classinfo(cid)

)

---5.向studentinfo表中插入3条数据

insert into studentinfo

values (14,'张三',1, '2020/2/3','12345678900','22@126.com', 2),

(100,'张三',0,'2020/2/3','12345678900','22@126.com', 2),

(109,'张三',1,'2020/2/3','12345678900','',1)

select \* from studentinfo

---------------------------------------------------触发器-------------------------------------------

use dbstu1

select \* from studentinfo

select \* from classinfo

--1. 创建一个日志表classinfo\_log,记录对班级信息表的操作.logid,opdate,opuser,opname

create table classinfo\_log

(

logid int identity (1,1) primary key,---操作序号

opdate datetime default getdate(),------操作日期时间

opuser nvarchar(50),--------------------操作者

opname nvarchar(50)----------------------操作名

)

select \* from classinfo\_log

--2.创建一个触发器tg\_class,当对班级信息表进行增加记录时,将修改记录到 classinfo\_log中

----drop trigger tg\_class

create trigger tg\_class

on classinfo after insert

as

insert into classinfo\_log(opuser,opname)

select user\_name(),'增'--------将操作人和操作名记录到日志文件中

from inserted;

--3.班级信息表中增加一条记录,查看日志表classinfo\_log中记录,验证触发器tg\_class是否启动

insert into classinfo

values(3)

select \* from classinfo\_log

--4.创建一个触发器tg2\_class,当对班级信息表进行修改数据时,将修改记录到 classinfo\_log中

---drop trigger tg2\_class

create trigger tg2\_class

on classinfo after update

as

insert into classinfo\_log(opuser,opname)

select user\_name(),'改'

from inserted;

--5. 班级信息表中修改一条记录,查看日志表classinfo\_log中记录,验证触发器tg2\_class是否启动

update classinfo

set ctitle='shuzimeiti'

where cid=3

select \* from classinfo\_log

----6.创建触发器t2,当对学生表执行 update操作时，该触发器被触发，不允许修改学生表中的姓名

use dbstu1

select \* from studentinfo

----------------------

--drop trigger t2

create trigger t2

on studentinfo

for update

as

if (update(sname))

begin

print'事物不能被处理，基本数据不能修改！'

rollback transaction

end

else

print'数据修改成功！'

-------7.修改学生表中sname，验证触发器是否被触发

update studentinfo

set sname='12'

where sid=4

select \* from studentinfo