CS21数据库设计-实验报告



指导老师： 曹卫东、黄超

姓 名： 韦阿辉

学 号： 2017122061

日 期： 2017年 11月

目 录

[1 数据定义语言DLL实验 1](#_Toc497836767)

[1.1 实验目的 1](#_Toc497836768)

[1.2 实验内容与步骤 1](#_Toc497836769)

[2 数据操作语言DML实验 6](#_Toc497836770)

[2.1 实验目的 6](#_Toc497836771)

[2.2 实验内容与步骤 6](#_Toc497836772)

[3 视图操作实验 15](#_Toc497836773)

[3.1 实验目的 15](#_Toc497836774)

[3.2 实验内容与步骤 15](#_Toc497836775)

[收获与感悟 19](#_Toc497836776)

[致 谢 19](#_Toc497836777)

[附录A：数据定义语言DLL实验代码 20](#_Toc497836778)

[附录B：数据操作语言DML实验代码 22](#_Toc497836779)

[附录C：视图操作实验代码 28](#_Toc497836780)

# 数据定义语言DLL实验

## 实验目的

1、熟悉 ORACLE 的数据库管理和数据库操作环境。

2、通过 SQL 的 DDL 定义和修改基本表。

3、通过 SQL 的 DDL 建立与删除索引。

## 实验内容与步骤

1. **进入ORACLE的环境，创建scott用户，至少分配CONNECT和RESOURCE角色,使用户拥有一定的数据库操作权限。**

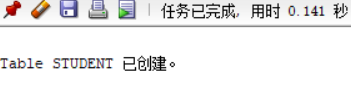
用户名：scott

口 令：tiger

1. **在SQL命令环境中用create table语句创建学生表（Student）、课程表（Course）和学生选课表（SC）三个基本表。**
2. 例如定义Student基本表，其中Sno为主码。

CREATE TABLE student

(SNO CHAR(5) NOT NULL PRIMARY KEY,

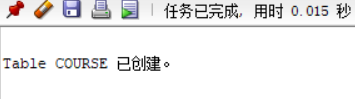
SNAME VARCHAR2(10) UNIQUE,

SSEX CHAR(2) NOT NULL,

SAGE NUMBER(3) DEFAULT 0,

SDEPT VARCHAR2(10));

1. 定义Course基本表;

CREATE TABLE Course

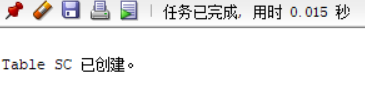
(CNO CHAR(10) NOT NULL,

CNAME VARCHAR2(15) NOT NULL,

CPNO CHAR(10),

CCREDIT NUMBER(10));

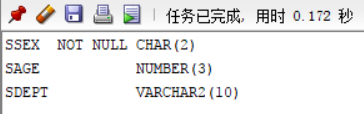
1. 定义SC基本表;

CREATE TABLE SC

(SNO CHAR(5) NOT NULL,

CNO CHAR(10) NOT NULL,

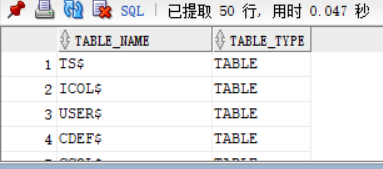
GRADE NUMBER(3));



1. 显示基本表结构:

DESC 基本表名:

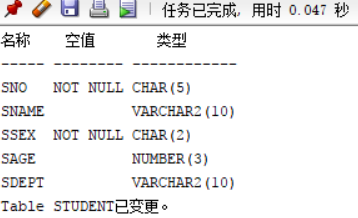
例如：DESC Student;

DESC student;

1. 显示scott用户模式内的基本表和视图;

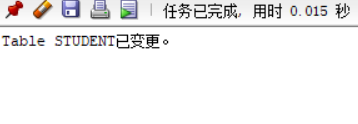
select \* from cat；

1. **用Alter Table 语句修改基本表结构;**
2. 对Student表增加“籍贯”属性列，VARCHAR2(20);

alter table student

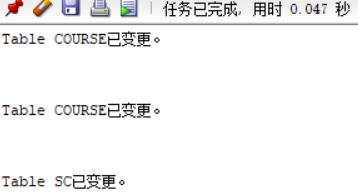
add jiguan varchar2(20);

1. 对Student表增加“入学时间”属性列，日期型;

alter table student

add timeschool date;

1. 对Course和SC表更新涉及“课程号”，“先行课号”属性列长度为2;

alter table Course

modify CNO CHAR(2);

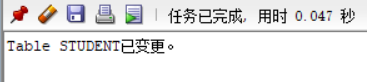
alter table Course

modify CPNO CHAR(2);

alter table SC

modify CNO CHAR(2);

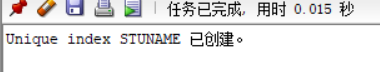
1. 删除学生姓名必须取惟一值的约束。

alter table student

DROP UNIQUE(SNAME);

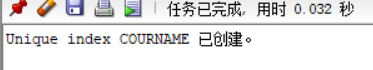
1. **用Create Index语句建立索引**
2. 对Studednt表按姓名升序建唯一索引;

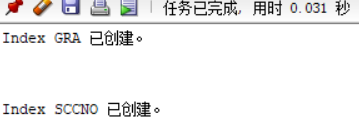
CREATE UNIQUE INDEX STUNAME ON student (sname ASC);



1. 对Course表按课程名称升序建唯一索引;

CREATE UNIQUE INDEX COURNAME ON Course (CNAME ASC);



1. 对SC表按成绩降序和学号升序建立索引;

CREATE INDEX GRA ON SC(GRADE DESC);

CREATE INDEX SCCNO ON SC(CNO ASC);

1. **可以用Drop语句删除基本表及索引:**

注意保存建立基本表或索引的命令，以便重新生成。

1. **用Insert语句向以上三个基本表各插入不少于十个元组**

例如： insert into student values('95001','李勇','男',20,'cs','北京', ' 27-2月 -05');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s1', 'sn1', 'm', '19', 'cs');

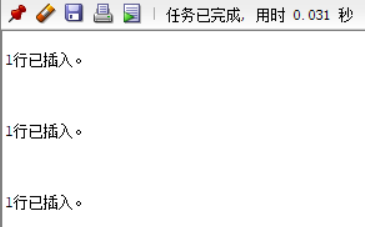
INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s2', 'sn2', 'f', '20', 'cs');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s3', 'sn3', 'm', '20', 'is');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s4', 'sn4', 'm', '10', 'is');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s5', 'sn5', 'f', '22', 'en');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s6', 'ssn6', 'm', '23', 'is');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s7', 'ssn7', 'f', '22', 'cs');

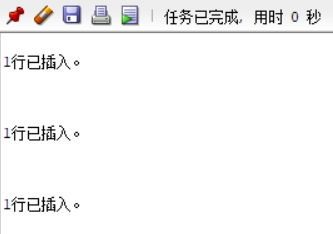
INSERT INTO COURSE (CNO, CNAME, CCREDIT) VALUES ('c01', 'cn1', '2');

INSERT INTO COURSE (CNO, CNAME, CPNO, CCREDIT) VALUES ('c02', 'cn2', 'c01', '1');

INSERT INTO COURSE (CNO, CNAME, CPNO, CCREDIT) VALUES ('c03', 'cn3', 'c01', '2');

INSERT INTO COURSE (CNO, CNAME, CPNO, CCREDIT) VALUES ('c04', 'cn4', 'c02', '1.5');

INSERT INTO COURSE (CNO, CNAME, CPNO, CCREDIT) VALUES ('c05', 'cn5', 'c01', '2.5');



INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s1', 'c01', '80');

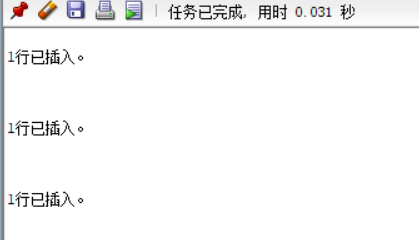
INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s1', 'c02', '85');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s2', 'c01', '90');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s2', 'c03', '88');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s3', 'c02', '89');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s3', 'c03', '90');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s4', 'c01', '92');

1. **用简单Select语句显示各表的内容;**

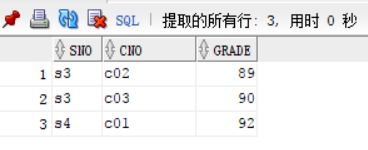
例如：Select \* from Student; Select sysdate from dual; --显示系统时间.

SELECT \* FROM Student;

SELECT \* FROM Course;

SELECT \* FROM SC;





# 数据操作语言DML实验

## 实验目的

1、 建立基本表并进行 DML 操作。

2、 数据查询：单表查询、连接查询、嵌套查询、集合查询和统计等功能。

3、 数据更新：增加、删除、查询和修改功能。

## 实验内容与步骤

1. **进入ORACLE的环境，进入scott用户模式**
2. **用 Create、Insert 语句建立基本表 S、P、J 和 SPJ**

其中：

1. S (SNO，SNAME，STATUS，CITY)

S表示供应商表，SNO为供应商代号，SNAME为供应商名字，STATUS为供应商状态，CITY为供应商所在城市，主关键字为SNO。

1. P (PNO，PNAME，COLOR，WEIGHT)

P表示零件表，PNO为零件代号，PNAME为零件名称，COLOR为零件颜色，WEIGHT为零件重量，主关键字为PNO。

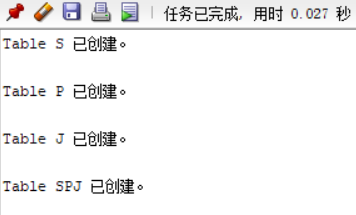
1. J (JNO，JNAME, CITY)

J表示工程表，JNO为工程编号，JNAME为工程名字，CITY为工程所在城市，主关键字为JN。

1. SPJ (SNO，PNO，JNO，QTY)

SPJ表示供应关系表，SNO是为指定工程提供零件的供应商代号，PNO为所提供的 零件代号，JNO为工程编号，QTY表示提供的零件数量，主关键字为SNO，PNO，JNO，

外关键字为 SNO，PNO，JNO。

create table S (SNO char(3) primary key,

SNAME char(10),

STATUS char(2),

CITY char(10));

create table P (PNO char(3) primary key,

PNAME char(10),

COLOR char(4),

WEIGHT int

);

create table J ( JNO char(3) primary key,

JNAME char(10),

CITY char(10));

create table SPJ (SNO char(3),

PNO char(3),

JNO char(3),

QTY int,

primary key (SNO,PNO,JNO),

foreign key (SNO) references S (SNO),

foreign key (PNO) references P (PNO),

foreign key (JNO) references J (JNO));

insert into S values('S1','精益','20','天津');

insert into S values('S2','盛锡','10','北京');

insert into S values('S3','东方红','30','北京');

insert into S values('S4','丰泰盛','30','天津');

insert into S values('S5','为民','30','上海');

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

insert into P values('P2','螺楦','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 J values('J1','三建','北京');

insert into J values('J2','一汽','长春');

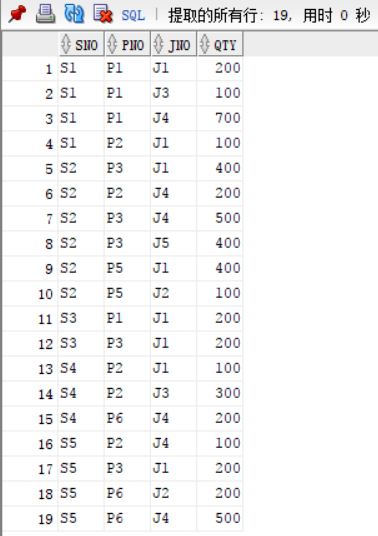
insert into J values('J3','弹簧厂','天津');

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','常州');

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

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','J1','100');

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

insert into SPJ values('S2','P2','J4','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('S3','P1','J1','200');

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

insert into SPJ values('S4','P2','J1','100');

insert into SPJ values('S4','P2','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','J3','200');

1. **用 Select 语句对上述四个基本表进行如下查询**
2. 求供应工程J1零件的供应商号码SNO；

--3-1

select SNO

from SPJ

where JNO = 'J1';

1. 求供应工程J1零件P1的供应商号码SNO；

--3-2

select SNO

from SPJ

where JNO = 'J1' AND PNO = 'P1';

1. 求供应工程J1零件为红色的供应商号码SNO；

--3-3

select SPJ.SNO

from SPJ,P

WHERE SPJ.PNO = P.PNO AND JNO = 'J1' AND COLOR = '红';

1. 求没有使用天津供应商生产的红色零件的工程号JNO；

--3-4

select DISTINCT JNO

from P,S,SPJ

WHERE SPJ.SNO = S.SNO AND

SPJ.PNO = P.PNO AND

P.COLOR != '红' AND

S.CITY != '天津';

1. 求至少用了供应商S1所供应的全部零件的工程号；

--3-5

select JNO

FROM SPJ

WHERE SNO = 'S3'

GROUP BY JNO

HAVING COUNT(\*)>=6;

1. 找出所有供应商的姓名和所在城市；

--3-6

select SNAME,CITY

FROM S;

1. 找出所有零件的名称、颜色、重量；

--3-7

select PNAME,COLOR,WEIGHT

FROM P;

1. 找出使用供应商S1所供应零件的工程号；

--3-8

select DISTINCT JNO

FROM SPJ

WHERE SNO = 'S1';

1. 找出工程项目J2使用的各种零件的名称及其数量；

--3-9

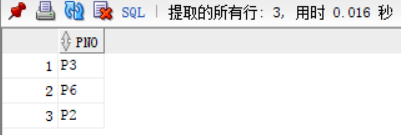
select P.PNAME,SPJ.QTY

FROM SPJ,P

WHERE SPJ.PNO = P.PNO AND

JNO = 'J2';

1. 找出上海厂商供应的所有零件号码；

--3-10

select DISTINCT PNO

FROM S,SPJ

WHERE S.SNO = SPJ.SNO AND

S.CITY = '上海';

1. **用Insert、Delete和Update语句实现如下数据更新；**
2. 将全部红色零件的颜色改成蓝色;

update p

set color='蓝'

where COLOR='红';

commit work;

1. 将工程J3的城市改为上海;

update J

set CITY='上海'

where JNO='J3';

commit work;

1. 由S5供给J4的零件P6改为由S3供应;

update SPJ

set SNO='S3'

where SNO='S5' and PNO='P6';

commit work;

1. 从供应商关系中删除S2的元组，并从供应情况关系中删除相应元组； （注意元组删除顺序）;

delete from SPJ

where SNO='S2';

delete from S

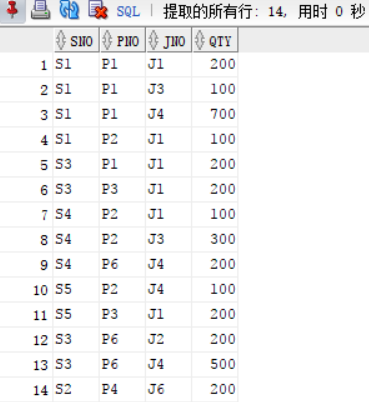
where SNO='S2';

1. 请将S2向工程项目J6供应200个P4零件的信息加入到供应关系;

insert into S values('S2', '盛锡',10，'北京');

--由于前一问中已经将S2从供应商表S中删除，

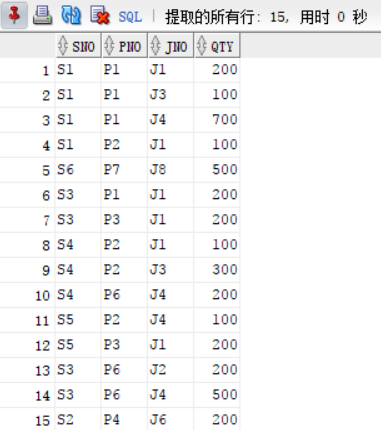
因此，需要在此处将其重新插入。

insert into SPJ values('S2','P4','J6',200);

1. 请将S6向工程项目J8供应500个P7零件的信息加入到供应关系； （注意涉及到几个表的添加元组操作？表的添加顺序如何？）

insert into S values('S6',NULL, NULL, NULL);

insert into P values('P7',NULL, NULL, NULL);

insert into J values('J8', NULL, NULL);

insert into SPJ values('S6','P7','J8', 500);

1. **注：以下操作利用学生关系模式Student、Course和SC三个基本表进行.**

用Insert、Delete和Update语句实现如下数据更新。

1. 对每一个系，求学生的平均年龄，并把结果存入数据库。 提示：首先在数据库中建立一个新表，用其中一列存放系名称，另一列存放该系学生的平均年龄。然后对Student表按系分组求平均年龄，并将系名和平均年龄存入新表中。

create table AveAge --创建新表

(ADEPT VARCHAR2(10),

AVAGE NUMBER(3) DEFAULT 0

);

select AVG(SAGE)

from student

where SDEPT='cs'

group by SDEPT;

select AVG(SAGE)

from student

where SDEPT='is'

group by SDEPT;

select AVG(SAGE)

from student

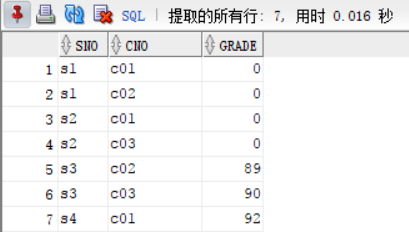
where SDEPT='en'

group by SDEPT;

insert into AveAge values('cs',20.33 );

insert into AveAge values('is',17.67);

insert into AveAge values('en',20.00);

1. 将计算机科学系全体学生的成绩置零。

update SC

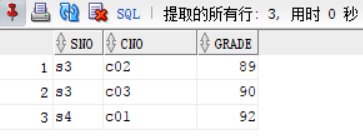
set GRADE=0

where SNO in (select SNO

from STUDENT

where SDEPT='cs');

1. 删除计算机科学系所有学生的选课记录。

delete from SC

where SNO IN

(select SNO

from STUDENT

WHERE SDEPT='cs'

);

1. 删除某学生的信息及其选课信息。（注：参照完整性，级联删除）

delete from SC where SNO = 's1';

delete from student where SNO='s1';



# 视图操作实验

## 实验目的

1. 理解视图的作用，能够根据实际需求用SQL创建视图；
2. 对基本表按需定义视图，并理解可以通过视图实现对基本表的操作。

## 实验内容与步骤

1. **进入ORACLE的环境，进入scott用户模式**

用户名： scott

口 令： tiger

（注：以下操作利用学生关系模式Student、Course和SC三个基本表进行）

1. **用Create view语句定义视图**
2. 建立信息系学生的视图,然后向其中插入一计算机系学生数据，分别查看视图和基本表;

create view INFORMATION(SNO, SNAME, SSEX, SAGE, SDEPT)

as

select \*

from STUDENT where SDEPT = 'is';

select \* from INFORMATION;

insert into INFORMATION (SNO, SNAME, SSEX, SAGE, SDEPT) values ('s7', 'ssn7', 'f', '22', 'cs');

1. 建立信息系学生的视图，并要求进行修改和插入操作时仍需保证该视图只有信息系的学生（Create view带有with check option）,然后向其中插入一计算机系学生数据，比较（1），（2）的不同；

create view INFORMATION2(SNO, SNAME, SSEX, SAGE, SDEPT)

as

select \*

from STUDENT where SDEPT = 'is'

with check option;

select \* from INFORMATION2;

insert into INFORMATION2 (SNO, SNAME, SSEX, SAGE, SDEPT) values ('s7', 'ssn7', 'f', '22', 'cs');

1. 建立信息系选修了1号课程且成绩在90分以上的学生的视图;

create view ISCSB

as

select student.SNO, student.SNAME, student.SSEX, student.SAGE, student.SDEPT,sc.CNO,sc.GRADE

from student,sc

where student.sno = sc.sno and sc.cno = 'c01' and sc.GRADE >= 90 and sc.CNO = 'is';

select \* from ISCSB;



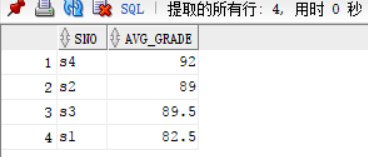
1. 将学生的学号及他的平均成绩定义为一个视图S-G;

create view SG(sno,avg\_grade)

as

select sno,avg(grade)

from sc

group by sno;

select \* from SG;

1. 将SC中成绩在相应课程平均成绩之上的元组定义成一个视图GOOD-SC。

create view GOODSC

as

select sno,cno,grade

from sc

where grade >

(select avg(grade)

from sc);

1. **通过视图对基本表进行操作**
2. 在信息系学生的视图中找出年龄小于20岁的学生;

select \*

from INFORMATION

where SAGE < 20;

1. 查询信息系选修了1号课程且成绩在90分以上的学生;

select \*

from ISCSB

where grade >= 90;

1. 在S-G视图中查询平均成绩在90分以上的学生学号和平均成绩;

select \*

from SG

where avg\_grade>=90;

1. 创建CS-KC视图，包括计算机专业各学生的学号、其选修的课程号及成绩,

要保证对该视图的修改都要符合专业名为“计算机”这个条件；

1. 查找计算机专业的学生学号和选修的课程号;
2. 向S-G视图中插入一元组。（操作结果如何？请解释。）

***说明：视图正如其名称，主要用来进行数据查询而不是更新。工程中多利用视图从不同角度，灵活、方便的查询一个或多个基本表、视图中的相关数据。***

收获与感悟

通过本门课程，我对数据结构有了基本的了解，这为我在编写其他计算机语言时带来了极大的便利。此外，我掌握了基础的SQL server语言，通过习题课的训练，我能够进行一些基本的操作，例如：数据定义语言DLL、数据操作语言DML、视图操、触发器、范式等。

致 谢

在这里尤其要感谢曹卫东老师和黄超老师，两位老师讲课逻辑性很强，课堂内的例子生动形象，通俗易懂，使得我能很快的吸收课堂内的知识。黄老师对学生很是体贴，耐心地解答了我所有的疑问。感谢百度、CSDN上的各位博主，他们的博客对我帮助很多。

# 附录A：数据定义语言DLL实验代码

drop table student;

drop table Course;

drop table SC;

--2.1

CREATE TABLE student

(SNO CHAR(5) NOT NULL PRIMARY KEY,

SNAME VARCHAR2(10) UNIQUE,

SSEX CHAR(2) NOT NULL,

SAGE NUMBER(3) DEFAULT 0,

SDEPT VARCHAR2(10));

--student

--(SNO,SNAME,SSEX,SAGE,SDEPT);

--2.2

CREATE TABLE Course

(CNO CHAR(10) NOT NULL,

CNAME VARCHAR2(15) NOT NULL,

CPNO CHAR(10),

CCREDIT NUMBER(1));

--2.3

CREATE TABLE SC

(SNO CHAR(5) NOT NULL,

CNO CHAR(10) NOT NULL,

GRADE NUMBER(3));

--2.4

DESC student;

--2.5

select \* from cat;

--3.1

alter table student

add jiguan varchar2(20);

--3.2

alter table student

add timeschool date;

--3.3

alter table Course

modify CNO CHAR(2);

alter table Course

modify CPNO CHAR(2);

alter table SC

modify CNO CHAR(2);

--3.4

alter table student

DROP UNIQUE(SNAME);

--4

CREATE UNIQUE INDEX STUNAME ON student (sname ASC);

CREATE UNIQUE INDEX COURNAME ON Course (CNAME ASC);

CREATE INDEX GRA ON SC(GRADE DESC);

CREATE INDEX SCCNO ON SC(CNO ASC);

--5

--6

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s1', 'sn1', 'm', '19', 'cs');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s2', 'sn2', 'f', '20', 'cs');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s3', 'sn3', 'm', '20', 'is');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s4', 'sn4', 'm', '10', 'is');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s5', 'sn5', 'f', '22', 'en');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s6', 'ssn6', 'm', '23', 'is');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s7', 'ssn7', 'f', '22', 'cs');

INSERT INTO COURSE (CNO, CNAME, CPNO, CCREDIT) VALUES ('c01', 'cn1','c02',1);

INSERT INTO COURSE (CNO, CNAME, CPNO, CCREDIT) VALUES ('c02', 'cn2', 'c01', '1');

INSERT INTO COURSE (CNO, CNAME, CPNO, CCREDIT) VALUES ('c03', 'cn3', 'c01', '2');

INSERT INTO COURSE (CNO, CNAME, CPNO, CCREDIT) VALUES ('c04', 'cn4', 'c02', '1.5');

INSERT INTO COURSE (CNO, CNAME, CPNO, CCREDIT) VALUES ('c05', 'cn5', 'c01', '2.5');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s1', 'c01', '80');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s1', 'c02', '85');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s2', 'c01', '90');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s2', 'c03', '88');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s3', 'c02', '89');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s3', 'c03', '90');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s4', 'c01', '92');

--7.1

select \* from student;

--7.2

select \* from course;

--7.3

select \* from sc;

# 附录B：数据操作语言DML实验代码

DROP TABLE S CASCADE CONSTRAINT;

DROP TABLE P CASCADE CONSTRAINT;

DROP TABLE J CASCADE CONSTRAINT;

DROP TABLE SPJ CASCADE CONSTRAINT;

create table S

(

SNO char(3) primary key,

SNAME char(10),

STATUS char(2),

CITY char(10)

);

create table P

(

PNO char(3) primary key,

PNAME char(10),

COLOR char(4),

WEIGHT int

);

create table J

(

JNO char(3) primary key,

JNAME char(10),

CITY char(10)

);

create table SPJ

(

SNO char(3),

PNO char(3),

JNO char(3),

QTY int,

primary key (SNO,PNO,JNO),

foreign key (SNO) references S (SNO),

foreign key (PNO) references P (PNO),

foreign key (JNO) references J (JNO)

);

insert into S values('S1','精益','20','天津');

insert into S values('S2','盛锡','10','北京');

insert into S values('S3','东方红','30','北京');

insert into S values('S4','丰泰盛','30','天津');

insert into S values('S5','为民','30','上海');

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

insert into P values('P2','螺楦','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 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','常州');

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

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','J1','100');

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

insert into SPJ values('S2','P2','J4','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('S3','P1','J1','200');

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

insert into SPJ values('S4','P2','J1','100');

insert into SPJ values('S4','P2','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');

--3-1

select SNO

from SPJ

where JNO = 'J1';

--3-2

select SNO

from SPJ

where JNO = 'J1' AND PNO = 'P1';

--3-3

select SPJ.SNO

from SPJ,P

WHERE SPJ.PNO = P.PNO AND JNO = 'J1' AND COLOR = '红';

--3-4

select DISTINCT JNO

from P,S,SPJ

WHERE SPJ.SNO = S.SNO AND

SPJ.PNO = P.PNO AND

P.COLOR != '红' AND

S.CITY != '天津';

--3-5

select JNO

FROM SPJ

WHERE SNO = 'S3'

GROUP BY JNO

HAVING COUNT(\*)>=6;

--3-6

select SNAME,CITY

FROM S;

--3-7

select PNAME,COLOR,WEIGHT

FROM P;

--3-8

select DISTINCT JNO

FROM SPJ

WHERE SNO = 'S1';

--3-9

select P.PNAME,SPJ.QTY

FROM SPJ,P

WHERE SPJ.PNO = P.PNO AND

JNO = 'J2';

--3-10

select DISTINCT PNO

FROM S,SPJ

WHERE S.SNO = SPJ.SNO AND

S.CITY = '上海';

--3-11

select DISTINCT J.JNAME

FROM S,SPJ,J

WHERE S.SNO = SPJ.SNO AND J.JNO = SPJ.JNO AND

S.CITY = '上海';

--3-12

select DISTINCT SPJ.JNO

FROM S,SPJ

WHERE S.SNO = SPJ.SNO AND

S.CITY != '天津';

--3-13

select S.SNO,S.SNAME,P.PNO,P.PNAME,J.JNO,J.JNAME,SPJ.QTY

FROM S,P,J,SPJ

WHERE S.SNO = SPJ.SNO AND P.PNO = SPJ.PNO AND J.JNO = SPJ.JNO;

--3-14

select SNO,PNO,SUM(QTY)

FROM SPJ

GROUP BY SNO,PNO;

--4.1将全部红色零件的颜色改为蓝色

update p

set color='蓝'

where COLOR='红';

commit work;

select \*

from p;

--4.2将工程J3的城市改为上海

update J

set CITY='上海'

where JNO='J3';

commit work;

select \*

from J;

--4.3由S5供给J4的零件P6改为由S3供应

update SPJ

set SNO='S3'

where SNO='S5' and PNO='P6';

commit work;

select \*

from SPJ;

--4.4从供应商关系中删除S2的元组，并从供应情况关系中删除相应元组(注意原则删除顺序)；

delete from SPJ

where SNO='S2';

delete from S

where SNO='S2';

select \*

from S;

--4.5请将S2向工程项目J6供应200个P4零件的信息加入到供应关系中；

insert into S values('S2', '盛锡',10，'北京'); --由于前一问中已经将S2从供应商表S中删除，因此，需要在此处将其重新插入。

insert into SPJ values('S2','P4','J6',200);

select \*

from SPJ;

--4.6请将S6向工程项目J8供应500个P7零件的信息加入到供应关系。涉及到向4个表中添加原色操作，顺序如下。

insert into S values('S6',NULL, NULL, NULL);

insert into P values('P7',NULL, NULL, NULL);

insert into J values('J8', NULL, NULL);

insert into SPJ values('S6','P7','J8', 500);

commit work;

select \*

from SPJ;

--TP2-5.1对每一个系，求学生的平均年龄，并把结果存入数据库

create table AveAge --创建新表

(ADEPT VARCHAR2(10),

AVAGE NUMBER(3) DEFAULT 0

);

select AVG(SAGE)

from student

where SDEPT='cs'

group by SDEPT;

select AVG(SAGE)

from student

where SDEPT='is'

group by SDEPT;

select AVG(SAGE)

from student

where SDEPT='en'

group by SDEPT;

insert into AveAge values('cs',20.33 );

insert into AveAge values('is',17.67);

insert into AveAge values('en',20.00);

select \*

from AveAge;

--TP2-5.2将计算机科学系全体学生的成绩置0

update SC

set GRADE=0

where SNO in (select SNO from STUDENT where SDEPT='cs');

select \*

from SC;

--TP2-5.3删除计算机科学系所有学生的选课记录

delete from SC

where SNO IN

(select SNO

from STUDENT

WHERE SDEPT='cs'

);

--TP2-5.4删除某学生的信息及其选课信息。(注：参照完整性，级联删除)

delete from SC where SNO = 's1';

delete from student where SNO='s1';

# 附录C：视图操作实验代码

drop table student;

drop table Course;

drop table sc;

create table student(

sno char(5) primary key,

sname varchar(10) unique,

ssex char(2) not null,

sage number(3) default 0,

sdept varchar(10));

create table Course(

cno char(3) primary key,

cname varchar2(15) not null,

cpno char(3),

ccredit number(1));

create table sc(

sno char(5),

cno char(3) ,

grade number(3) check (grade<=100 and grade>=0),

primary key(sno,cno),

CONSTRAINT FK\_sno

FOREIGN KEY (sno)

REFERENCES student(sno),

CONSTRAINT FK\_cno

FOREIGN KEY (cno)

REFERENCES course(cno)

);

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s1', 'sn1', 'm', '19', 'cs');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s2', 'sn2', 'f', '20', 'cs');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s3', 'sn3', 'm', '20', 'is');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s4', 'sn4', 'm', '10', 'is');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s5', 'sn5', 'f', '22', 'en');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s6', 'ssn6', 'm', '23', 'is');

INSERT INTO STUDENT (SNO, SNAME, SSEX, SAGE, SDEPT) VALUES ('s7', 'ssn7', 'f', '22', 'cs');

INSERT INTO COURSE (CNO, CNAME, CCREDIT) VALUES ('c01', 'cn1', '2');

INSERT INTO COURSE (CNO, CNAME, CPNO, CCREDIT) VALUES ('c02', 'cn2', 'c01', '1');

INSERT INTO COURSE (CNO, CNAME, CPNO, CCREDIT) VALUES ('c03', 'cn3', 'c01', '2');

INSERT INTO COURSE (CNO, CNAME, CPNO, CCREDIT) VALUES ('c04', 'cn4', 'c02', '1.5');

INSERT INTO COURSE (CNO, CNAME, CPNO, CCREDIT) VALUES ('c05', 'cn5', 'c01', '2.5');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s1', 'c01', '80');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s1', 'c02', '85');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s2', 'c01', '90');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s2', 'c03', '88');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s3', 'c02', '89');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s3', 'c03', '90');

INSERT INTO SC (SNO, CNO, GRADE) VALUES ('s4', 'c01', '92');

--2-1

create view INFORMATION(SNO, SNAME, SSEX, SAGE, SDEPT)

as

select \*

from STUDENT where SDEPT = 'is';

select \* from INFORMATION;

insert into INFORMATION (SNO, SNAME, SSEX, SAGE, SDEPT) values ('s7', 'ssn7', 'f', '22', 'cs');

--2-2

create view INFORMATION2(SNO, SNAME, SSEX, SAGE, SDEPT)

as

select \*

from STUDENT where SDEPT = 'is'

with check option;

select \* from INFORMATION2;

insert into INFORMATION2 (SNO, SNAME, SSEX, SAGE, SDEPT) values ('s7', 'ssn7', 'f', '22', 'cs');

--2-3

create view ISCSB

as

select student.SNO, student.SNAME, student.SSEX, student.SAGE, student.SDEPT,sc.CNO,sc.GRADE

from student,sc

where student.sno = sc.sno and sc.cno = 'c01' and sc.GRADE >= 90 and sc.CNO = 'is';

select \* from ISCSB;

--2-4

create view SG(sno,avg\_grade)

as

select sno,avg(grade)

from sc

group by sno;

select \* from SG;

--2-5

create view GOODSC

as

select sno,cno,grade

from sc

where grade >

(select avg(grade)

from sc);

--3-1

select \*

from INFORMATION

where SAGE < 20;

--3-2

select \*

from ISCSB

where grade >= 90;

--3.3

select \*

from SG

where avg\_grade>=90;

--3.4

create view CS\_KC

as