电子信息学院

实验报告书

课程名： 大型数据库

题 目： 实验七 操作视图

实验类别 【实践】

班 级： 软件1712

学 号： 171003530214

姓 名： 张飒

评语：

实验态度：认真（ ） 一般（ √ ） 较差（ ）

实验结果：正确（ ） 部分正确（√ ）错（ ）

实验理论： 掌握（ √ ） 熟悉（ ） 了解（ ） 生疏（ ）

操作技能：较强（ ） 一般（ √ ） 较差（ ）

实验报告： 较好（ ） 一般（ √ ） 较差（ ）

成绩： 指导教师： 林志杰

批阅时间：2019年 10月 21 日

评语：

实验态度：认真（ ） 一般（ ） 较差（ ）

实验结果：正确（ ） 部分正确（ ）错（ ）

实验理论：掌握（ ） 熟悉（ ） 了解（ ） 生疏（ ）

操作技能：较强（ ） 一般（ ） 较差（ ）

实验报告：较好（ ） 一般（ ） 较差（ ）

成绩： 指导教师：

批阅时间： 年 月 日

评语：

实验态度：认真（ ） 一般（ ） 较差（ ）

实验结果：正确（ ） 部分正确（ ）错（ ）

实验理论：掌握（ ） 熟悉（ ） 了解（ ） 生疏（ ）

操作技能：较强（ ） 一般（ ） 较差（ ）

实验报告：较好（ ） 一般（ ） 较差（ ）

成绩： 指导教师：

批阅时间： 年 月 日

1．实验目的

（1）掌握oracle 11g的SQL语言中的视图的相关操作。

（2）熟练oracle创建数据库的过程，能够对视图进行深入理解和掌握。

（3）掌握oracle的中视图的存在的意义和作用。

（4）掌握SQL语言中操作视图的基本技能。

2．实验目标：

在本次实验中，学生应该能够利用所学视图的基本理论知识，进行视图的创建、更新、删除和修改视图的基本操作，能够在实际应用中熟练运用视图。

视图的作用：

视图的作用实在是太强大了，以下是视图的好处：

【作用一】

    提高了重用性，视图就像一个函数。如果要频繁获取user的name和goods的name。就应该使用以下sql语言。如下：

        select a.name as username, b.name as goodsname from user as a, goods as b, ug as c where a.id=c.userid and c.goodsid=b.id;

    但有了视图就不一样了，创建视图other。示例：

        create view other as select a.name as username, b.name as goodsname from user as a, goods as b, ug as c where a.id=c.userid and c.goodsid=b.id;

    创建好视图后，就可以这样获取user的name和goods的name。如下：

        select \* from other;

    以上sql语句，就能获取user的name和goods的name了。

【作用二】

    对数据库重构，却不影响程序的运行。假如因为某种需求，需要将user拆成表usera和表userb，该两张表的结构如下：

       usera有id，name，age字段

       userb有id，name，sex字段

    这时如果使用sql语句：select \* from user;那就会提示该表不存在，这时该如何解决呢。解决方案：创建视图。以下sql语句创建视图：

        create view user as select a.name,a.age,b.sex from usera as a, userb as b where a.name=b.name;

        以上假设name都是唯一的。使用sql语句：select \* from user;就不会报错什么的。这就实现了更改数据库结构，不更改脚本程序的功能了。

【作用三】

    提高了安全性能。可以对不同的用户，设定不同的视图。例如：某用户只能获取user表的name和age数据，不能获取sex数据。则可以这样创建视图。示例如下：

        create view other as select a.name, a.age from user as a;

    这样的话，使用sql语句：select \* from other; 最多就只能获取name和age的数据，其他的数据就获取不了了。

【作用四】

    让数据更加清晰。想要什么样的数据，就创建什么样的视图。

简单总结视图的作用：

1、视图能够简化用户的操作

2、视图使用户能以多种角度看待同一数据

3、视图对重构数据库提供了一定程度的逻辑独立性

4、视图能够对机密数据提供安全保护

5、适当的利用视图可以更清晰的表达查询

问题：哪类视图是可以更新的？哪类视图是不可更新的？各举一例说明。

答：基本表的行列子集视图一般是可更新的。若视图的属性来自集函数、表达式，则该视图肯定是不可以更新的。

（1）若视图的字段是来自字段表达式或常数，则不允许对此视图执行INSERT、UPDATE操作，允许执行DELETE操作；

（2）若视图的字段是来自库函数，则此视图不允许更新；

（3）若视图的定义中有GROUP BY子句或聚集函数时，则此视图不允许更新；

（4）若视图的定义中有DISTINCT任选项，则此视图不允许更新；

（5）若视图的定义中有嵌套查询，并且嵌套查询的FROM子句中涉及的表也是导出该视图的基表，则此视图不允许更新；

（6）若视图是由两个以上的基表导出的，此视图不允许更新；

（7）一个不允许更新的视图上定义的视图也不允许更新；

（8）由一个基表定义的视图，只含有基表的主键或候补键，并且视图中没有用表达式或函数定义的属性，才允许更新。

3．实验内容与步骤

注意：完成上述两个步骤后，而且必须进行上述数据库连接成功之后，才能进行后续的实验步骤，在连接成功的数据库窗口就可以直接写sql语句进行数据库的表操作了。

在做本次实验之前（结果截图贴在题目下方）：

创建S、SC和C三张表及添加表命令如下：

Create Table S

(Sno CHAR(5) NOT NULL,

Sname VARCHAR(20),

Sage SMALLINT CHECK(Sage>=15 AND Sage<=45),

Ssex CHAR(2),

Sdept CHAR(10),constraint pr\_sno primary key(sno));

Create Table C

(Cno CHAR(2) NOT NULL,

Cname VARCHAR(20) NOT NULL,

Cpno CHAR(2),

Ccredit SMALLINT,constraint pr\_Cno primary key(Cno));

Create Table SC(

Sno CHAR(5) NOT NULL,

Cno CHAR(2) NOT NULL,

Grade SMALLINT,constraint pr\_Cno primary key(Sno,Cno));

INSERT INTO Student VALUES(‘18001’,’钱横’,18,’男’,’CS’);

INSERT INTO Student VALUES (‘18002’,’王林’,18,’男’,’CS’);

INSERT INTO Student VALUES (‘18003’,’李民’,20,’男’,’IS’);

INSERT INTO Student VALUES (‘18004’,’赵蕊’,18,’女’,’MA’);

INSERT INTO Student VALUES (‘18005’,’姬彧’,19,’女’,’MA’);

INSERT INTO Student VALUES (‘18006’,’姬彧卿’,18,’女’,’CS’);

INSERT INTO Student VALUES (‘08007’,’姬彧瑾’,18,’女’,’CS’);

INSERT INTO Course VALUES (‘1’,’数据库系统’,’5’,4);

INSERT INTO Course VALUES (‘2’,’数学分析’,null,2);

INSERT INTO Course VALUES (‘3’,’信息系统导论’,’1’,3);

INSERT INTO Course VALUES (‘4’,’操作系统原理’,’6’,3);

INSERT INTO Course VALUES (‘5’,’数据结构’,’7’,4);

INSERT INTO Course VALUES (‘6’,’数据处理基础’,null,4);

INSERT INTO Course VALUES (‘7’,’C语言’,’6’,3);

INSERT INTO SC VALUES (‘18001’,’1’,87);

INSERT INTO SC VALUES (‘18001’,’3’,90);

INSERT INTO SC VALUES (‘18001’,’2’,67);

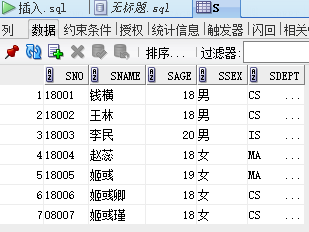
INSERT INTO SC VALUES (‘18002’,’2’,95);

INSERT INTO SC VALUES (‘18002’,’3’,88);

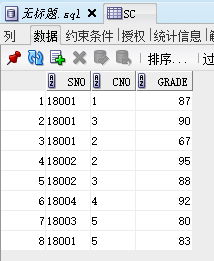
INSERT INTO SC VALUES (‘18004’,’4’,92);

INSERT INTO SC VALUES (‘18003’,’5’,80);

INSERT INTO SC VALUES (‘18001’,’5’,83);







1. 创建并添加各表记录后，可逐个实践如下各题，运行结果截图在相应题目下方：
2. 创建新视图view\_s,包含学生的学号、姓名和系别；

create view view\_s as

select sno,sname,sdept from s;

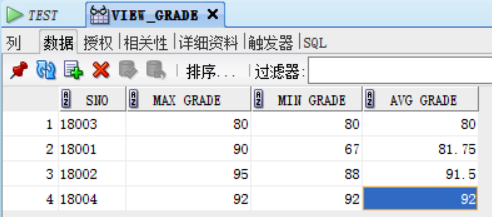


1. 创建新视图view\_grade，显示每个学生的成绩最高分、最低分和平均成绩；

create view view\_grade as

select sno,max(grade) as max\_grade,min(grade) as min\_grade,avg(grade) as avg\_grade

from sc group by sno;

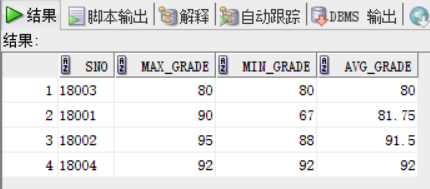


1. 查询上面建立的视图；

select \* from view\_s;



select \* from view\_grade;



1. 更新视图view\_s，把所有学好为08开头的学生的相关系别信息改为管理系MS；

update view\_s

set sdept='MS' where sno like '08%';



1. 更新视图view\_grade，插入记录(‘08091’,95,63,87)；

insert into view\_grade values('08091',95,63,87)

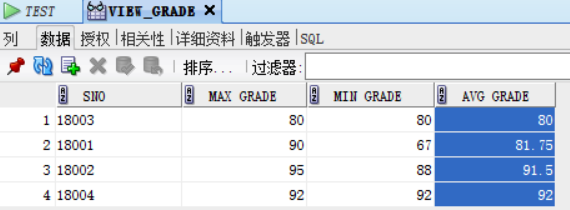


1. 创建新视图view\_grade，显示每个学生的成绩最高分、最低分和平均成绩，要求才视图只读；

create view view\_grade as

select sno,max(grade) as max\_grade,min(grade) as min\_grade,avg(grade) as avg\_grade

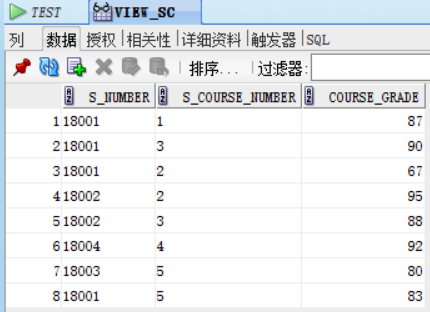
from sc group by sno with read only;



1. 在SC表上建立视图view\_sc，映射该表的全部列，要求在视图中列明分别为“s\_number”“s\_course\_number”“course\_grade”；

create view view\_sc as

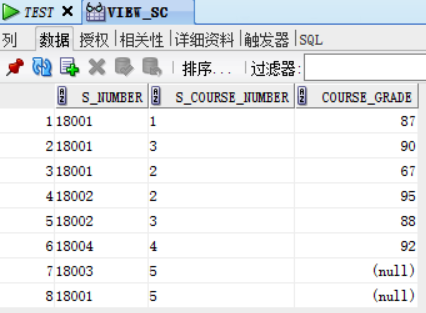
select sno s\_number,cno s\_course\_number, grade course\_grade from sc



1. 更新视图view\_sc，将所有学生的数据库成绩置空；

update view\_sc

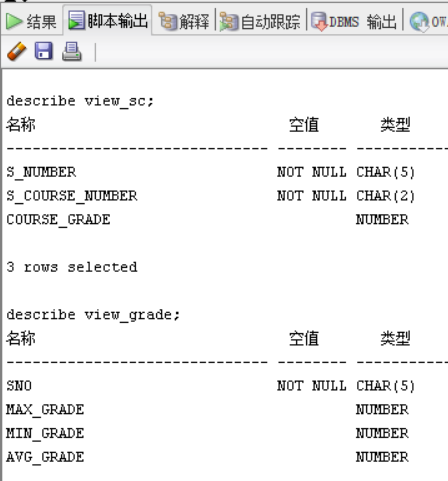
set course\_grade=null where s\_course\_number='5'



1. 查询一下view\_sc、view\_grade的结构；

describe view\_sc;

describe view\_grade;



1. 修改视图view\_s，使其显示所有男同学的信息，并且年龄按照从大到小排序；

create or replace view view\_s as

select \* from s where ssex='男' order by sage desc;



1. 查询视图view\_s中信息，将视图中插入两条男同学的信息，要求一个年龄19岁，一个20岁，学号、姓名等其他信息自己定义；

select \* from view\_s



insert into view\_s values('18008','张三',19,'男','CS');

insert into view\_s values('18009','李四',20,'男','CS');7



1. 将视图view\_s中所有小于19岁的学社给你的信息删除，查询视图中信息；

delete from view\_s where sage<19;

select \* from view\_s;



1. 删除视图view\_sc、view\_grade；

drop view view\_sc;

drop view view\_grade;



4.结果分析与实验体会