**本 科 实 验 报 告**

实验名称： 数据库第三次实验

|  |  |  |  |
| --- | --- | --- | --- |
| 学 员： | 于鹄杰 | 学 号： | 201506021075 |
| 培养类型： | 工程技术类 | 年 级： | 2015级 |
| 专 业： | 网络工程（军事物联网方向） | 所属学院： | 计算机学院 |
| 指导教员： | 周竞文 | 职 称： | 讲师 |

国防科技大学教务处制

**数据源（一）**

CREATE TABLE S(

Sno CHAR(9) PRIMARY KEY,

Sname CHAR(20) NOT NULL,

Sage NUMBER(3),

Ssex CHAR(2) CHECK(Ssex IN ('M','F')));

INSERT INTO S VALUES ('S1', 'Wang', '20', 'M');

INSERT INTO S VALUES ('S2', 'Liu', '19', 'M');

INSERT INTO S VALUES ('S3', 'Chen', '22', 'M');

INSERT INTO S VALUES ('S4', 'Wu', '19', 'M');

INSERT INTO S VALUES ('S5', 'Lou', '21', 'F');

INSERT INTO S VALUES ('S8', 'Dong', '18', 'F');

CREATE TABLE C(

Cno CHAR(9) PRIMARY KEY,

Cname CHAR(40) NOT NULL,

teacher CHAR(20));

INSERT INTO C VALUES ('C2', 'Maths', 'Ma');

INSERT INTO C VALUES ('C4', 'Physics', 'Shi');

INSERT INTO C VALUES ('C3', 'Chemistry', 'Zhou');

INSERT INTO C VALUES ('C1', 'DB', 'Li');

INSERT INTO C VALUES ('C5', 'OS', 'Wen');

CREATE TABLE SC(

Sno CHAR(9),

Cno CHAR(9),

grade NUMBER(3) CHECK(grade BETWEEN 0 AND 100),

PRIMARY KEY(Sno,Cno),

FOREIGN KEY(Sno) REFERENCES S(Sno),

FOREIGN KEY(Cno) REFERENCES C(Cno));

INSERT INTO SC VALUES ('S1', 'C1', 80);

INSERT INTO SC VALUES ('S1', 'C2', 70);

INSERT INTO SC VALUES ('S1', 'C3', 85);

INSERT INTO SC VALUES ('S1', 'C4', 90);

INSERT INTO SC VALUES ('S1', 'C5', 70);

INSERT INTO SC VALUES ('S2', 'C1', 85);

INSERT INTO SC VALUES ('S2', 'C2', NULL);

INSERT INTO SC VALUES ('S2', 'C4', NULL);

INSERT INTO SC VALUES ('S3', 'C1', 90);

INSERT INTO SC VALUES ('S3', 'C2', 85);

INSERT INTO SC VALUES ('S3', 'C3', 95);

INSERT INTO SC VALUES ('S4', 'C1', 75);

INSERT INTO SC VALUES ('S4', 'C3', NULL);

INSERT INTO SC VALUES ('S4', 'C4', 70);

INSERT INTO SC VALUES ('S5', 'C1', 70);

INSERT INTO SC VALUES ('S5', 'C2', 60);

INSERT INTO SC VALUES ('S5', 'C3', 80);

INSERT INTO SC VALUES ('S5', 'C5', 65);

INSERT INTO SC VALUES ('S8', 'C1', 90);

INSERT INTO SC VALUES ('S8', 'C2', NULL);

INSERT INTO SC VALUES ('S8', 'C3', 90);

INSERT INTO SC VALUES ('S8', 'C5', NULL);

**数据源（二）**

部门（编号，名称，地址）

CREATE TABLE Dept(

DeptNO NUMBER(3) PRIMARY KEY,

Dname VARCHAR2(14),

Location VARCHAR2(13));

INSERT INTO DEPT VALUES(10,'Accounting','New York');

INSERT INTO DEPT VALUES (20,'Research','Dallas');

INSERT INTO DEPT VALUES (30,'Sales','Chicago');

INSERT INTO DEPT VALUES (40,'Operations','Boston');

员工（工号，姓名，职位，上司，受雇日期，工资，奖金，所在部门编号）

CREATE TABLE Emp(

EmpNO NUMBER(4) PRIMARY KEY,

Ename VARCHAR2(10),

Job VARCHAR2(9),

Mgr VARCHAR2(9),

Hiredate DATE,

Sal NUMBER(7,2),

Comm NUMBER(7,2),

DeptNO NUMBER(3),

FOREIGN KEY(DeptNO) REFERENCES Dept(DeptNO));

INSERT INTO EMP VALUES(7369,'Smith','Clerk',7902,'17-12月-05',800,NULL,20);

INSERT INTO EMP VALUES (7499,'Allen','Salesman',7698,'20-2月-02',1600,300,30);

INSERT INTO EMP VALUES (7521,'Ward', 'Salesman',7698,'22-2月-96',1250,500,30);

INSERT INTO EMP VALUES (7566,'Jones','Manager',7839,'02-4月-09',2975,NULL,20);

INSERT INTO EMP VALUES (7654,'Martin', 'Salesman',7698,'28-9月-10',1250,1400,30);

INSERT INTO EMP VALUES (7698,'Blake', 'Manager',7839,'01-5月-08',2850,NULL,30);

INSERT INTO EMP VALUES (7782,'Clark', 'Manager',7839,'09-6月-05',2450,NULL,10);

INSERT INTO EMP VALUES (7788,'Scott','Analyst',7566,'19-4月-10',3000,NULL,20);

INSERT INTO EMP VALUES (7839,'King','President',NULL,'17-11月-11',5000,NULL,10);

INSERT INTO EMP VALUES (7844,'Turner', 'Salesman',7698,'08-9月-07',1500,0,30);

INSERT INTO EMP VALUES (7876,'Adams', 'Clerk',7788,'23-5月-04',1100,NULL,20);

INSERT INTO EMP VALUES (7900,'James', 'Clerk',7698,'03-12月-03',950,NULL,30);

INSERT INTO EMP VALUES (7902,'Ford', 'Analyst',7566,'03-12月-07',3000,NULL,20);

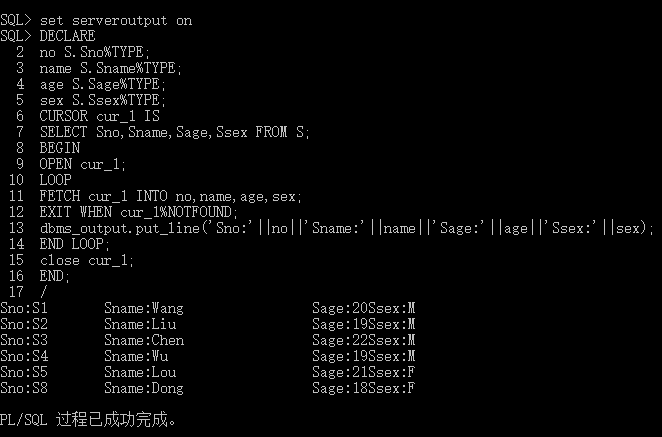
INSERT INTO EMP VALUES (7934,'Miller', 'Clerk',7782,'23-1月-09',1300,NULL,10);

1. **游标**
2. **游标查询**
3. 在**匿名块**中使用**显式游标**，查询基本表S的所有信息，并逐行打印。

对应的Oracle语句：

|  |
| --- |
| set serveroutput on  DECLARE  no S.Sno%TYPE;  name S.Sname%TYPE;  age S.Sage%TYPE;  sex S.Ssex%TYPE;  CURSOR cur\_1 IS  SELECT Sno,Sname,Sage,Ssex FROM S;  BEGIN  OPEN cur\_1;  LOOP  FETCH cur\_1 INTO no,name,age,sex;  EXIT WHEN cur\_1%NOTFOUND;  dbms\_output.put\_line('Sno:'||no||'Sname:'||name||'Sage:'||age||'Ssex:'||sex);  END LOOP;  close cur\_1;  END;  / |

截图结果为：

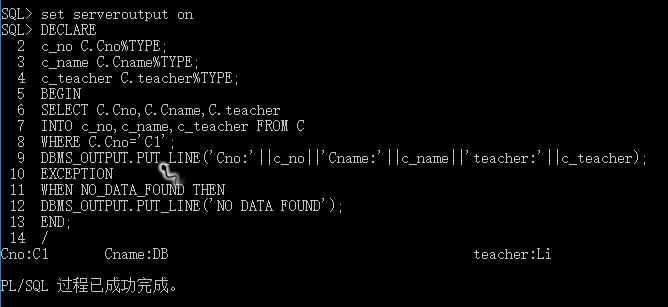


2）在**匿名块**中使用**隐式游标**，查询基本表C中课程号为C1的课程信息，并进行打印，要求进行异常处理（未找到C1课程如何处理）。

对应的Oracle语句：

|  |
| --- |
| set serveroutput on  DECLARE  c\_no C.Cno%TYPE;  c\_name C.Cname%TYPE;  c\_teacher C.teacher%TYPE;  BEGIN  SELECT C.Cno,C.Cname,C.teacher  INTO c\_no,c\_name,c\_teacher FROM C  WHERE C.Cno='C1';  DBMS\_OUTPUT.PUT\_LINE('Cno:'||c\_no||'Cname:'||c\_name||'teacher:'||c\_teacher);  EXCEPTION  WHEN NO\_DATA\_FOUND THEN  DBMS\_OUTPUT.PUT\_LINE('NO DATA FOUND');  END;  / |

截图结果为：

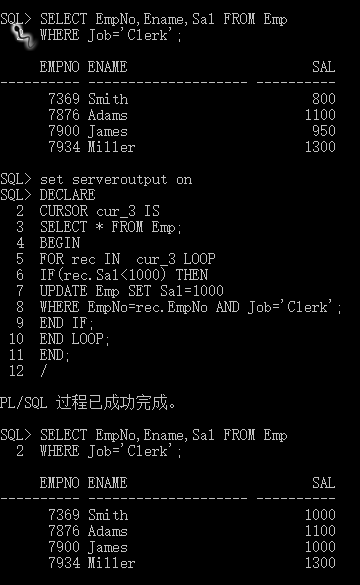


1. **游标修改**
2. 在**匿名块**中使用游标，对Emp表中职位为Clerk的职员的工资进行修改：凡工资低于1000的，调整为1000。

对应的Oracle语句：

|  |
| --- |
| set serveroutput on  DECLARE  CURSOR cur\_3 IS  SELECT \* FROM Emp;  BEGIN  FOR rec IN cur\_3 LOOP  IF(rec.Sal<1000) THEN  UPDATE Emp SET Sal=1000  WHERE EmpNo=rec.EmpNo AND Job='Clerk';  END IF;  END LOOP;  END;  /  SELECT EmpNo,Ename,Sal FROM Emp  WHERE Job='Clerk'; |

实验前后截图结果为：

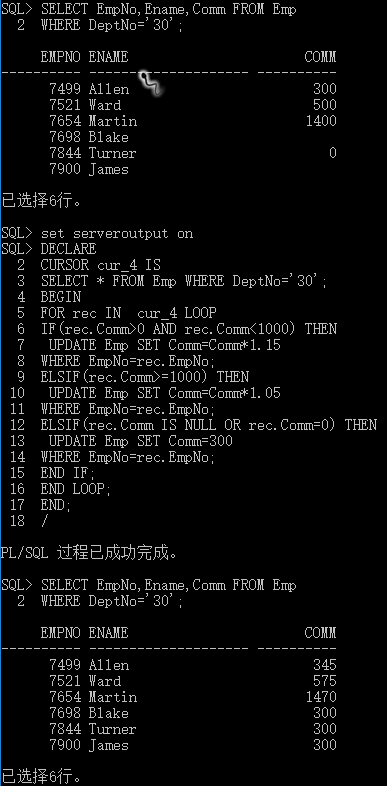


1. 在**匿名块**中使用游标，对Emp表中30号部门的职员的奖金进行调整：没有奖金的，调整为基本奖金300；奖金在1000以下的，提高15%；奖金在1000及以上的，提高5%。

对应的Oracle语句：

|  |
| --- |
| set serveroutput on  DECLARE  CURSOR cur\_4 IS  SELECT \* FROM Emp WHERE DeptNo='30';  BEGIN  FOR rec IN cur\_4 LOOP  IF(rec.Comm>0 AND rec.Comm<1000) THEN  UPDATE Emp SET Comm=Comm\*1.15  WHERE EmpNo=rec.EmpNo;  ELSIF(rec.Comm>=1000) THEN  UPDATE Emp SET Comm=Comm\*1.05  WHERE EmpNo=rec.EmpNo;  ELSIF(rec.Comm IS NULL OR rec.Comm=0) THEN  UPDATE Emp SET Comm=300  WHERE EmpNo=rec.EmpNo;  END IF;  END LOOP;  END;  /  SELECT EmpNo,Ename,Comm FROM Emp  WHERE DeptNo='30'; |

实验前后截图结果为：



1. **存储过程**

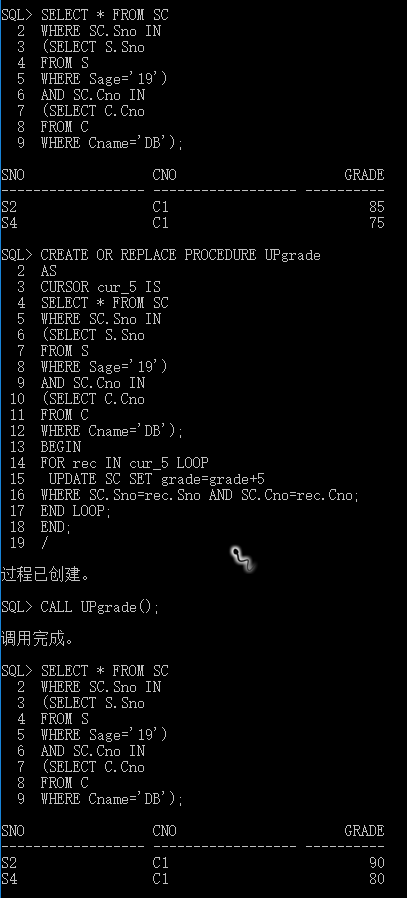
**1、不带差数的存储过程**

对基本表S、C和SC操作：用不带参数的PL/SQL存储过程将年龄为19岁的学生的DB课程成绩提高5分。

对应的Oracle语句：

|  |
| --- |
| CREATE OR REPLACE PROCEDURE UPgrade  AS  CURSOR cur\_5 IS  SELECT \* FROM SC  WHERE SC.Sno IN  (SELECT S.Sno  FROM S  WHERE Sage='19')  AND SC.Cno IN  (SELECT C.Cno  FROM C  WHERE Cname='DB');  BEGIN  FOR rec IN cur\_5 LOOP  UPDATE SC SET grade=grade+5  WHERE SC.Sno=rec.Sno AND SC.Cno=rec.Cno;  END LOOP;  END;  /  CALL UPgrade();  SELECT \* FROM SC  WHERE SC.Sno IN  (SELECT S.Sno  FROM S  WHERE Sage='19')  AND SC.Cno IN  (SELECT C.Cno  FROM C  WHERE Cname='DB'); |

实验前后截图结果为：



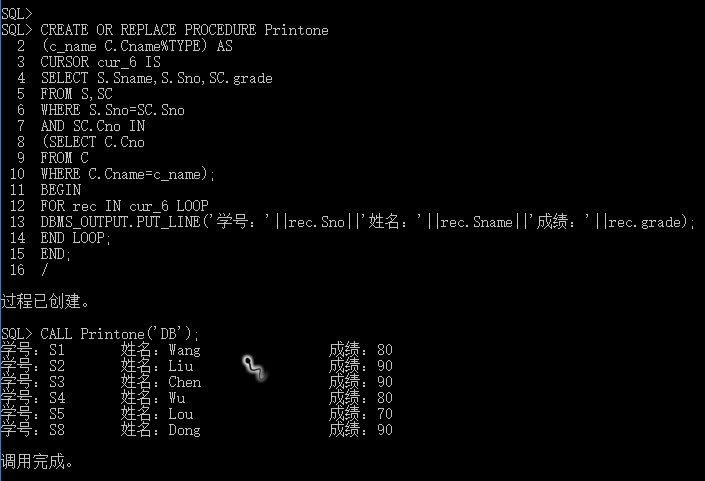
**2、带参数的存储过程**

用带参数的存储过程实现：对任意给定的课程名，查询并打印出所有选修该课程的学生的学号、姓名和成绩。

对应的Oracle语句：

|  |
| --- |
| CREATE OR REPLACE PROCEDURE Printone  (c\_name C.Cname%TYPE) AS  CURSOR cur\_6 IS  SELECT S.Sname,S.Sno,SC.grade  FROM S,SC  WHERE S.Sno=SC.Sno  AND SC.Cno IN  (SELECT C.Cno  FROM C  WHERE C.Cname=c\_name);  BEGIN  FOR rec IN cur\_6 LOOP  DBMS\_OUTPUT.PUT\_LINE('学号：'||rec.Sno||'姓名：'||rec.Sname||'成绩：'||rec.grade);  END LOOP;  END;  /  CALL Printone('DB'); |

实验前后截图结果为：



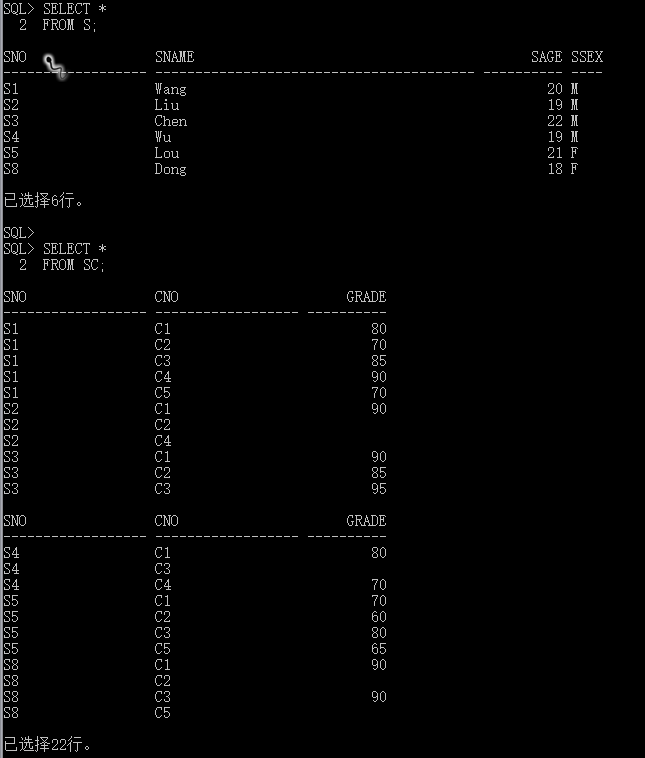
1. **触发器**
2. 生成一个基本表S的触发器，如果增加一个学生，则同时也在基本表SC中增加该同学的所有课程的初始成绩（初始成绩设置为60分）。

对应的Oracle语句：

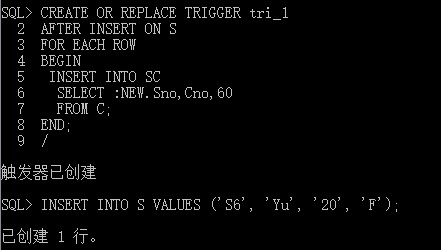
|  |
| --- |
| CREATE OR REPLACE TRIGGER tri\_1  AFTER INSERT ON S  FOR EACH ROW  BEGIN  INSERT INTO SC  SELECT :NEW.Sno,Cno,60  FROM C;  END;  /  INSERT INTO S VALUES ('S6', 'Yu', '20', 'F');  SELECT \*  FROM S;  SELECT \*  FROM SC; |

实验前后截图结果为：

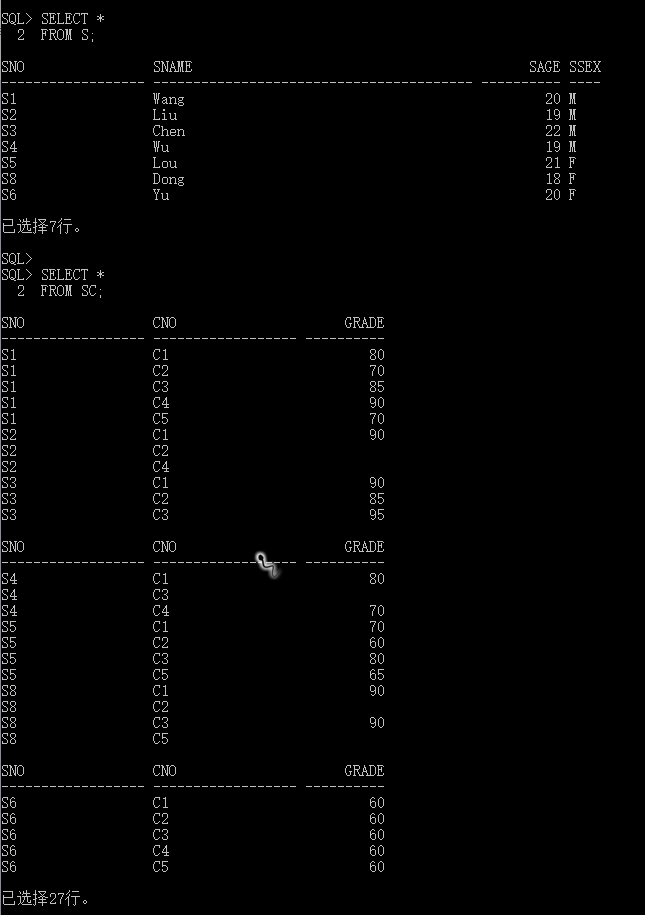
实验前：



实验中：



实验后：



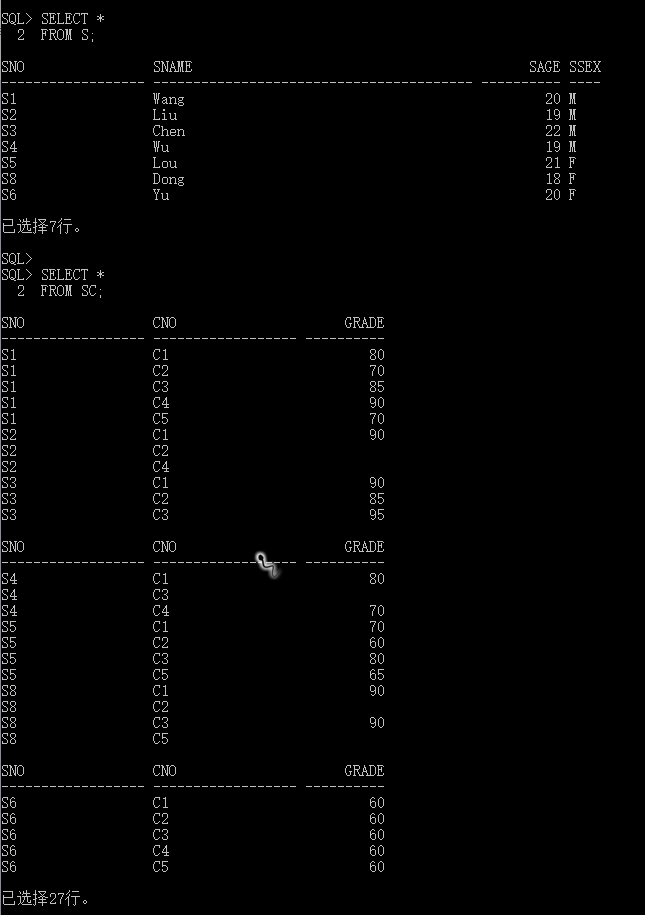
1. 生成一个基本表S的触发器，如果删除一个学生，则同时也在基本表SC中删除所有该同学的课程成绩。

对应的Oracle语句：

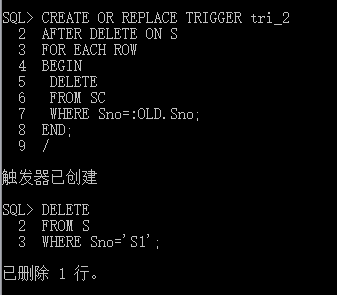
|  |
| --- |
| SELECT \*  FROM Emp; |

实验前后截图结果为：

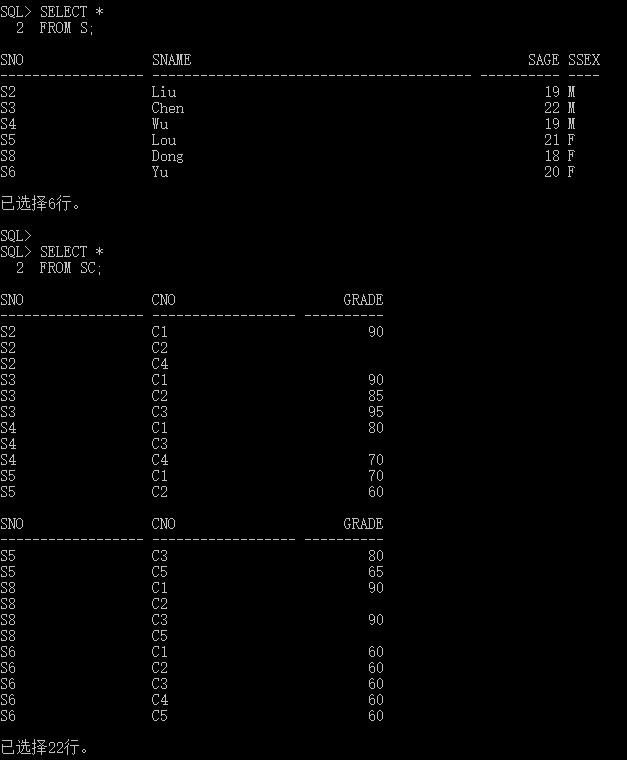
实验前：



实验中：



实验后：



1. 生成一个表Emp的触发器，增加一个职员时，如果其工资高于所在部门当前的平均工资，则将其修改为平均工资值。

对应的Oracle语句：

|  |
| --- |
| CREATE OR REPLACE TRIGGER tri\_3  BEFORE INSERT ON Emp  FOR EACH ROW  DECLARE avgsal Emp.Sal%TYPE;  BEGIN  SELECT AVG(Sal) INTO avgsal  FROM Emp  WHERE DeptNo=:NEW.DeptNo;  IF(:NEW.Sal>avgsal) THEN  :NEW.Sal:=avgsal;  END IF;  END;  /  SELECT EmpNo,Sal,DeptNo  FROM Emp;  INSERT INTO EMP VALUES(1111,'WANG','Clerk',7902,'17-12月-05',10000,NULL,20); |

实验前后截图结果为：

