DML 操作是指对数据库中表记录的操作，主要包括表记录的插入、更新、删除和查询、是开发人员日常使用最频繁的操作

1.数据的增加

语法格式：

插入记录

   语法： insert into tablename[(field1,field2, 、、、 )] values (values1,values2,values3 、、、 );

[例]往student表中插入一条信息，studentid='S01'，studentname='张三',能否成功？为什么

不能插入成功，因为在数据表中有七条数据，而插入的信息只有两条，还有5条没有数据，数据库系统直接拒绝插入数据。需要在数据表中约束条件

可以多条数据同时插入：

INSERT INTO table\_name(col\_name1,col\_name2,……,col\_namen)

VALUES(value1,value2,……,valuen),

……

(value 1,value2,……,valuen);

[例]：往course表中插入多条信息

Insert into course(courseid,coursename,bookname,period,credit)

Values(‘Dp030002’,’数据库组成原理’,’数据库’,80,4);

有时需要将一张表的数据插入到另一张表中，此项操作也可以使用INSERT关键字实现，语法形式为：

INSERT INTO table\_name1(table\_name1\_col\_list)

SELECT table\_name2\_col\_list FROM table\_name2;

[例子]复制student的表结构，并将现有的student表数据放进去

复制表：create table student\_copy1 like student

复制数据：INSERT INTO 新表 SELECT \* FROM 旧表

2.数据的修改

语法格式：

   更新 ( 更改 ) 记录

   语法：

update tablename set

field1=value1, field2=value2, field3=value3

where column\_name=valuen;

Set处放要修改的内容

Where 筛选出你要修改的数据行

[例]：将张三修改成李四，性别修改为女

update student #修改student表

set studentname='李四',sex='女' #学生姓名修改为张三

where studentname= '张三'; #挑选出张三所在的行

3.数据的删除

删除记录

   语法： delete from tablename [where column\_name=value]

[例]：删除St0109010001同学的信息[grade表中该同学的信息删除了吗，该如何去做]

delete from student

where studentid='St0109010001';

以上只删除了student表中该学生的信息，但是grade中的信息没有删除

[例子]：删除’St0109010004‘号同学的地址

#删除St0109010004同学的地址

update student #修改student表

set HomeAddr=null

where studentid= 'St0109010004';

4.简单查询

单表查询是指从一张数据表中查询所需要的数据

SELECT        \* | <字段列表> }

FROM <表1>, <表2>…

[WHERE <表达式>

[GROUP BY <group by definition>]

 [HAVING <expression> [{<operator> <expression>}…]]

[ORDER BY <order by definition>]

[LIMIT [<offset>,] <row count>]         ;

4.1 全部列查询

[例]查询student表中所有同学的全部信息

Select \*from student;

查询所有字段数据，是指从一张表中检索出所有记录，查询方式有两种，一种是使用通配符“\*”，另一种是列出所有字段名

SELECT \* FROM table\_name;

4.2 列筛选

[例]查询student表的学号和姓名列

Select studentid,studentname from student;

查询指定字段的数据，语法形式如下：

SELECT col\_name1[,col\_name2,……,col\_namen] FROM table\_name;

[课堂练习]查询course表中所有课程的课程名和对应学分

Select CourseName,credit from course;

4.3 列函数的使用

[例]查询每个同学的学号，姓名，年龄（以2010年为计算标准）

select studentid,studentname,2010-year(birth1) from student;

[例]查询每个同学的学号，课程号，成绩及其是否及格(60以下，差，60-80，良，80以上，优秀)

select studentid,courseid,grade,

case when grade<60 then '差'

when grade>=60 and grade<=80 then '良好'

else '优秀' end '成绩等级'

from grade;

case when 函数：

CASE

WHEN [value1] THEN [result1]

WHEN [value2] THEN [result2]

…ELSE [default] END 列名

[课堂练习]查询班级表中每个班级的编号以及该班级对应的系（如果是Dp01,输出计算机系，如果是Dp02，输出英语系)

select classid,departmentid,

case when departmentid='Dp01' then '计算机系'

else '英语系' end '系别'

from class;

4.4 别名

如何给列设置名字？

# 字段设置别名：

SELECT 字段1 [AS] 别名1, 字段2 [AS] 别名2 [, …] FROM 表名

# 表设置别名

SELECT 表别名.字段 [, …] FROM 表名 [AS] 表别名

4.5 去重

[例]有多少个同学选过课？

select distinct studentid from grade;

去除重复记录

distinct：

select distinct 列名 from 表

select选项：

* 默认值为All，表示保存所有查询到的记录。
* 当设置为DISTINCT时，表示去除重复记录，只保留一条。
* 当查询记录的字段有多个时，必须所有字段的值完全相同才被认为是重复记录。

4.4 聚合函数

[例]输出所有同学所有课程的平均成绩

select studentid,(sum(grade)/6) as grade\_sum from grade group by studentid;

COUNT()、SUM()、AVG()、MAX()、MIN()和GROUP\_CONCAT()函数中可以在参数前添加DISTINCT，表示对不重复的记录进行相关操作。COUNT()的参数设置为“\*”时，表示统计符合条件的所有记录（包含NULL）。

|  |  |
| --- | --- |
| **函数名** | **描述** |
| COUNT() | 返回参数字段的数量，不统计为NULL的记录 |
| SUM() | 返回参数字段之和 |
| AVG() | 返回参数字段的平均值 |
| MAX() | 返回参数字段的最大值 |
| MIN() | 返回参数字段的最小值 |

[例]查找student表中一共有多少位同学以及年龄最大的学生姓名

select count(studentid) as '学生人数' from student;

select min(birth1) as '年龄最大' from student;

#查找年龄最大的

select StudentName from student where birth1 in (select min(birth1) from student);

实验：

1.在数据表中增加以下内容(用sql语句插入)：

顾客表(Customer)

|  |  |  |  |
| --- | --- | --- | --- |
| Cid | Cname | Cphone | VIP |
| 95001 | 李勇 | 95105366 | 食品会员，家居会员 |
| 95002 | 刘晨 | 96118 | 日用品会员，食品会员 |
| 95003 | 王敏 | 96178 | 日用品会员 |

产品表（Product）

|  |  |  |
| --- | --- | --- |
| Pid | Pname | Pprice |
| 1 | 矿泉水 | 2 |
| 2 | 食用油 | 60 |
| 3 | 薯片 | 4.5 |

订单表(Order)（Cotime是timestamp数据类型)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OID | Cid | Cnumber | Cotime | Cdtime |
| 001 | 95001 | 50 | 实时 | 2020-12-20 |
| 002 | 95001 | 24 | 实时 | 2020-9-30 |
| 003 | 95003 | 600 | 实时 | 2020-9-23 |

订单明细(details）

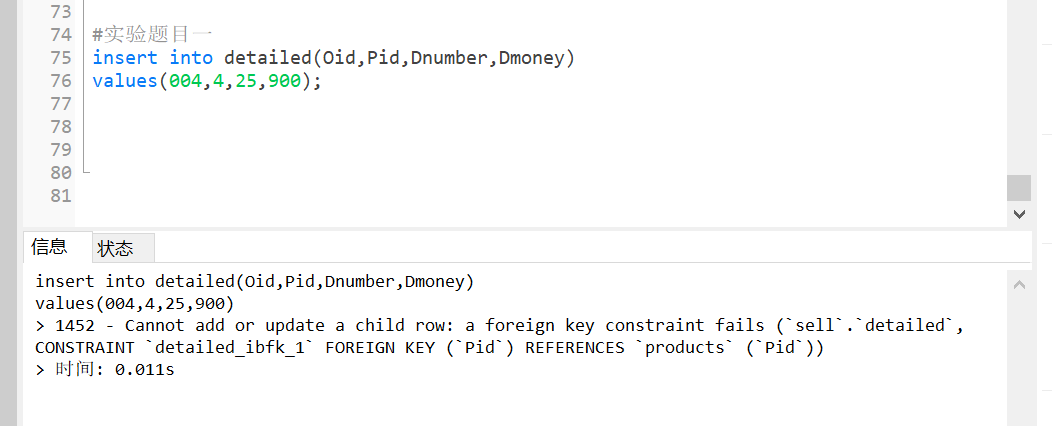
|  |  |  |  |
| --- | --- | --- | --- |
| OID | Pid | Dnumber | Dmoney |
| 001 | 1 | 25 |  |
| 001 | 2 | 25 |  |
| 002 | 3 | 24 |  |
| 003 | 1 | 100 |  |
| 003 | 2 | 150 |  |

1. 在订单明细表中插入（004,4,25,900）能不能成功？若能，请截图，若不能，为什么。

不能成功插入数据，因为订单明细表中的产品编号（PID）和产品表存在级联外键，而产品表中的产品编号（PID）是主键。

外键约束的条件就是：外键不是所属数据表的主键，但会对应着另外一张数据表的主键。

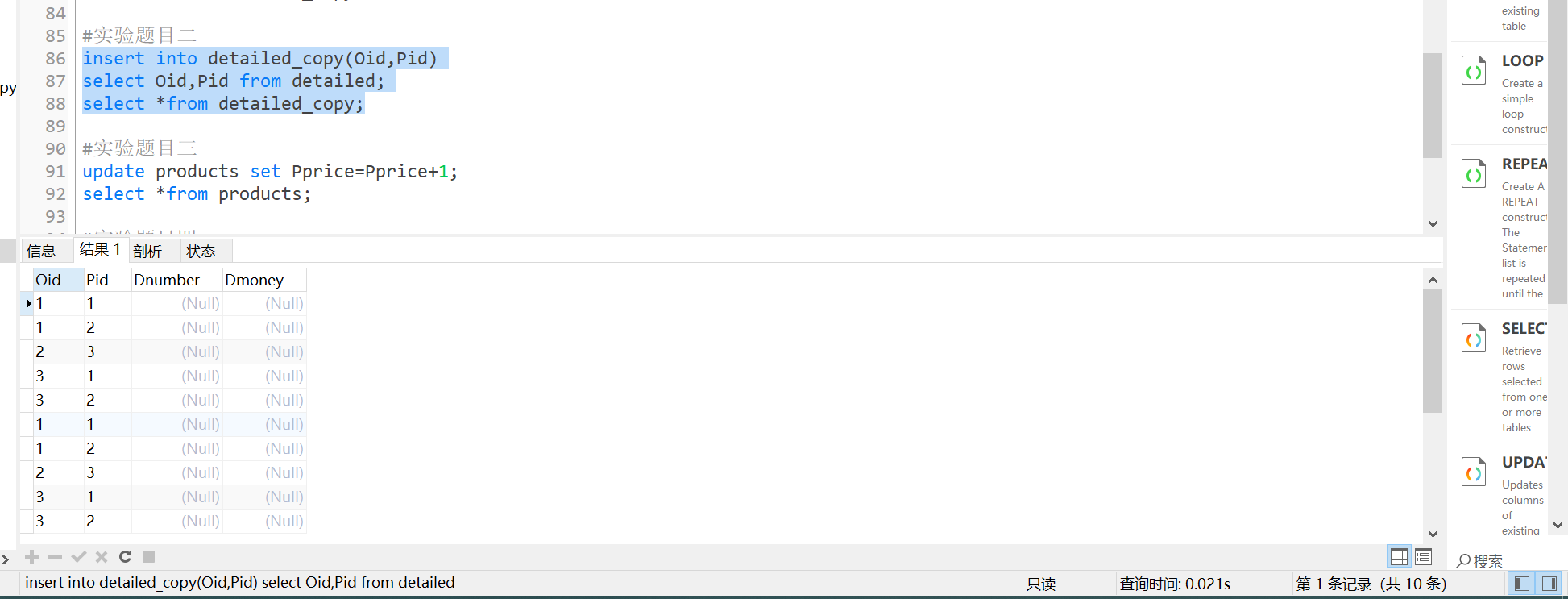
不能添加外键约束，外键约束中的pid引用了pid，不能添加或跟新数据



1. 建立表details\_copy，建立时套用details的表结构，通过获取details表的数据一次性插入OID和PID数据。

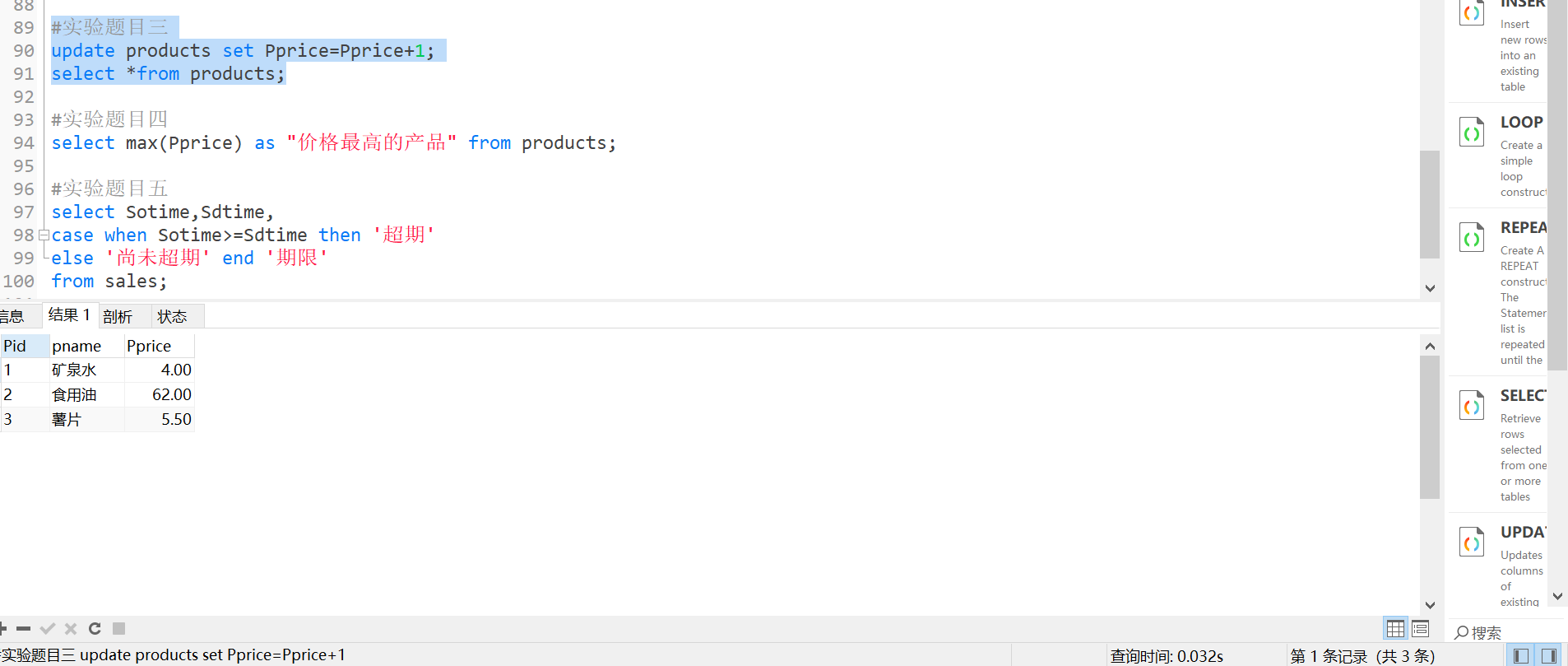
**insert into detailed\_copy(Oid,Pid)**

**select Oid,Pid from detailed;**



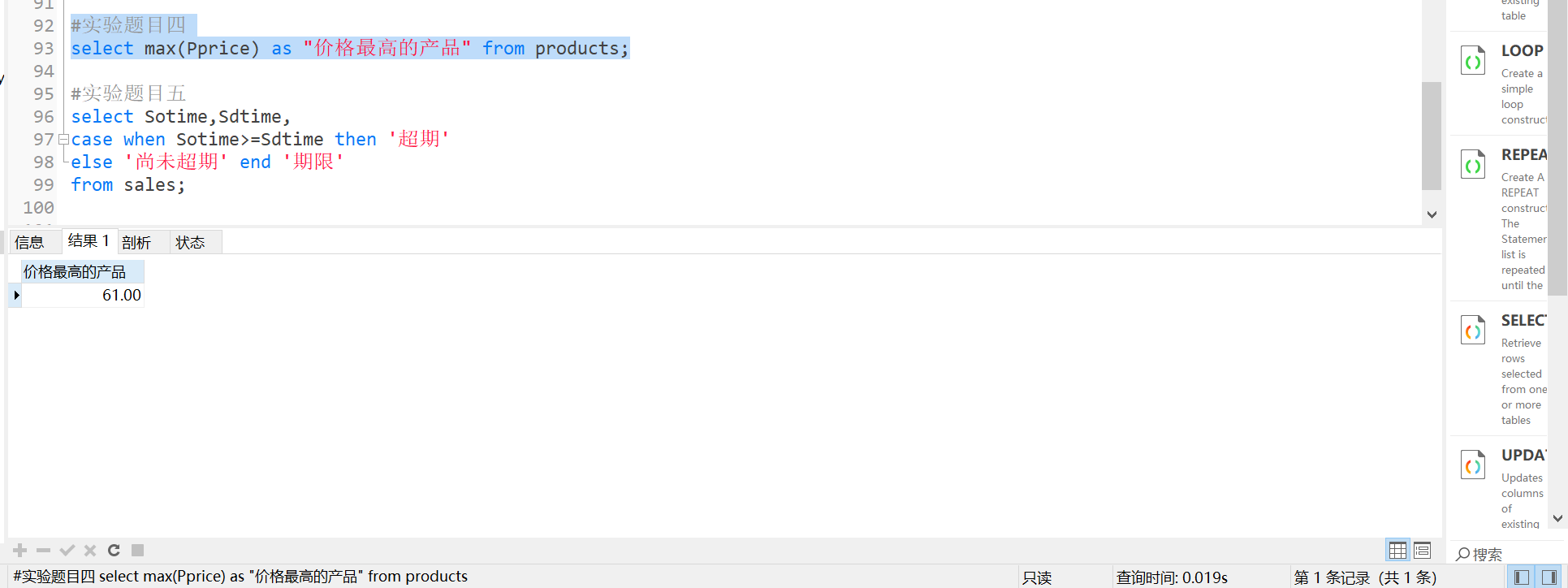
1. 将所有产品的价格在原来的基础上加1

**update products set Pprice=Pprice+1;**



1. 查找出价格最高的产品名称

**select max(Pprice) as "价格最高的产品" from products;**



1. 对比每个订单的cotime和cdtime，如果超期了，输出超期，如果未超期，输出 尚未超期

**select Sotime,Sdtime,**

**case when Sotime>=Sdtime then '超期'**

**else '尚未超期' end '期限'**

**from sales;**



6.输出每个同学姓名及所在省份

select StudentName,HomeAddr from student;

