

《数据库概论》实验二高级SQL 实验报告

姓名: 傅小龙学号: 191220029 联系方式: 1830970417(QQ)

一、实验环境

操作系统: Windows 10

Mysql Workbench 8.0 CE

mysql-8.0.19-winx64

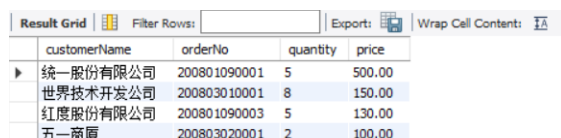
Visual Studio 2019(Community)

二、实验过程

任务1-(1):

```
1 drop procedure if exists gen_pruduct_sell_list;  
2 delimiter //  
3 create procedure gen_pruduct_sell_list(in pName Varchar(40))  
4 begin  
5     select C.customerName, O.orderNo, OD.quantity, OD.price  
6     from Product P, Customer C, ordermaster O, orderdetail OD  
7     where P.productNo = OD.productNo and C.customerNo = O.customerNo and O  
8         .orderNo = OD.orderNo and P.productName = pName  
9     order by OD.price desc;  
10 end;  
11 //  
12 call gen_pruduct_sell_list('32MLDRAM');
```

运行结果详见图1.



customerName	orderNo	quantity	price
统一股份有限公司	200801090001	5	500.00
世界技术开发公司	200803010001	8	150.00
红度股份有限公司	200801090003	5	130.00
五一商厦	200803020001	2	100.00

图 1: 1-(1)运行结果

任务1-(2):

```

1 drop procedure if exists earlierHired_employee;
2 delimiter //
3 create procedure earlierHired_employee(in eNumber Char(8))
4 begin
5     select E.employeeNo, E.employeeName, E.gender, E.hireDate, E.
6         department
7     from employee E
8     where E.hireDate < (
9         select E.hireDate
10        from employee E
11        where E.employeeNo = eNumber
12    ) and E.department = (
13        select E.department
14        from employee E
15        where E.employeeNo = eNumber
16    );
17 end
18 //
19 call earlierHired_employee('E2008005');

```

运行结果详见图2.

Result Grid					
Filter Rows:					
Export:					
Wrap Cell Content:					
	employeeNo	employeeName	gender	hireDate	department
▶	E2005001	喻自强	M	1990-02-06 00:00:00	财务科

图 2: 1-(2)运行结果

任务2-(1):

```

1 drop function if exists avg_order_price;
2 delimiter //
3 create function avg_order_price(pName Varchar(40))
4 returns float
5 DETERMINISTIC
6 begin
7     declare avg_price float;
8     select avg(OD.price) into avg_price
9     from orderdetail OD, Product P
10    where OD.productNo = P.productNo and P.productName = pName
11    group by P.productName;
12    return avg_price;
13 end;
14 //
15 select P.productName, avg_order_price(P.productName)
16 from Product P;

```

运行结果详见图3.

Result Grid			Filter Rows:		E
	productName	avg_order_price(P.productName)			
▶	32M DRAM	220			
	17寸显示器	350			
	120GB硬盘	220			
	3.5寸软驱	425			
	键盘	375			
	VGA显示卡	325			
	网卡	265			
	Pentium 100CPU	285			
	1G DDR	293.2			
	52倍速光驱	410			
	计算机字典	283.333			
	9600bits/s调制...	340			
	Pentium主板	220			
	硕泰克SL-K8...	265			
	龙基777FT纯...	244			

图 3: 2-(1)运行结果

任务2-(2):

```

1  drop function if exists sum_product_sell;
2  delimiter //
3  create function sum_product_sell(pNo char(9))
4  returns integer
5  DETERMINISTIC
6  begin
7      declare sum_sell integer;
8      select sum(OD.quantity) into sum_sell
9      from orderdetail OD
10     where OD.productNo = pNo
11     group by OD.productNo;
12     return sum_sell;
13 end;
14 //
15 select P.productNo, P.productName, sum_product_sell(P.productNo)
16 from Product P
17 where sum_product_sell(P.productNo) > 4;

```

运行结果详见图4.

Result Grid			Filter Rows:		Export:	Wrap Cell
	productNo	productName	sum_product_sell(P.productNo)			
▶	P20050001	32M DRAM	20			
	P20050003	120GB硬盘	8			
	P20050004	3.5寸软驱	7			
	P20050005	键盘	6			
	P20060002	网卡	7			
	P20060003	Pentium 100CPU	12			
	P20070001	1G DDR	15			
	P20070002	52倍速光驱	12			
	P20070003	计算机字典	11			
	P20070004	9600bits/s调制解调	5			
	P20080001	Pentium主板	7			
	P20080002	硕泰克SL-K8AN-R...	7			

图 4: 2-(2)运行结果

任务3-(1):

```

1 drop trigger if exists pPrice_insert;
2 delimiter //
3 create trigger pPrice_insert before insert on product
4     for each row
5 begin
6     if new.productPrice > 1000 then
7         set new.productPrice = 1000;
8     end if;
9 end;

```

采用如下sql语句进行测试, 运行结果详见图5.可以看到编号为'P20090002'的产品价格被修改为了1000.

```

1 insert into product
2 (productNo, productName, productClass, productPrice)
3 VALUES
4 ('P20090001', 'test1', 'test', 100),
5 ('P20090002', 'test2', 'test', 12000);
6 select * from product where productNo like 'P2009%';
7 delete from product where productNo like 'P2009%';

```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
productNo	productName	productClass	productPrice
P20090001	test1	test	100.00
P20090002	test2	test	1000.00

图 5: 3-(1)运行结果

任务3-(2):

```

1 drop trigger if exists raise_salary;
2 delimiter //
3 create trigger raise_salary before insert on ordermaster
4     for each row
5 begin
6     declare old_salary float;
7     set old_salary = (
8         select salary
9         from employee
10        where employeeNo = new.employeeNo
11    );
12    if (
13        select hireDate
14        from employee
15        where employeeNo = new.employeeNo
16    ) < date('1992-01-01') then

```

```

17         update employee set salary = old_salary * 1.08 where
           employeeNo = new.employeeNo;
18     else
19         update employee set salary = old_salary * 1.05 where
           employeeNo = new.employeeNo;
20     end if;
21 end

```

采用如下sql语句进行测试, 运行结果详见图6.可以看到1992年前入职的'E2021002'员工在完成一个订单后工资增长了8%, 1992年后入职的'E2021001'员工在完成一个订单后工资增长了5%, 而没有完成订单的'E2021003'员工工资没有增长.

```

1  insert into employee
2  (employeeNo, employeeName, gender, birthday, address, telephone, hireDate,
   department, headShip, salary)
3  VALUES
4  ('E2021001', 'test1', 'M', '1968-01-06', 'test', NULL, '1992-02-28', 'test
   ', 'test', 1000),
5  ('E2021002', 'test2', 'M', '1968-01-06', 'test', NULL, '1991-02-28', 'test
   ', 'test', 1000),
6  ('E2021003', 'test2', 'M', '1968-01-06', 'test', NULL, '1991-02-28', 'test
   ', 'test', 1000);
7  insert into ordermaster
8  (orderNo, customerNo, employeeNo, orderDate, orderSum, invoiceNo)
9  VALUES
10 ('P20210001', 'C20050001', 'E2021001', '2008-01-09', 0, 'I202100001'),
11 ('P20210002', 'C20050001', 'E2021002', '2008-01-09', 0, 'I202100001');
12
13 select employeeNo, hireDate, salary
14 from employee
15 where employeeNo like 'E2021%';
16 delete from ordermaster where orderNo like '2021%';
17 SET FOREIGN_KEY_CHECKS = 0;
18 delete from employee where employeeNo like 'E2021%';
19 SET FOREIGN_KEY_CHECKS = 1;

```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	employeeNo	hireDate	salary		
▶	E2021001	1992-02-28 00:00:00	1050.00		
	E2021002	1991-02-28 00:00:00	1080.00		
	E2021003	1991-02-28 00:00:00	1000.00		

图 6: 3-(2)运行结果

以下内容均采用C++与mySql建立连接:

```

1  const char user[] = "admin";
2  const char pswd[] = "123";
3  const char host[] = "localhost";

```

```

4      const char db[] = "orderdb";
5      unsigned int port = 3309;
6
7      MYSQL myConnect;
8      MYSQL_RES* result = NULL;
9      MYSQL_ROW sql_row;
10     int res = 0;
11     mysql_init(&myConnect);
12     if (mysql_real_connect(&myConnect, host, user, pswd, db, port, NULL,
13                             0)){
14         ...
15     }

```

任务4-(1): (C++)

```

1  char order_4_1[] = "select employeeNo, employeeName, salary from employee_
   order by salary desc limit 20";
2      res = mysql_query(&myConnect, order_4_1);
3      if (!res)
4      {
5          result = mysql_store_result(&myConnect);
6          if (result)
7          {
8              int num_fields = mysql_num_fields(result);
9              MYSQL_FIELD* field;
10             char space[] = "                ";
11             while (field = mysql_fetch_field(result)) {
12                 char s[128];
13                 sprintf(s, "%s%s", field->name, space);
14                 printf("%-20.20s", s);
15             }
16             cout << endl;
17             while (sql_row = mysql_fetch_row(result))
18             {
19                 for (int i = 0; i < num_fields; i++) {
20                     char s[128];
21                     sprintf(s, "%s%s", sql_row[i], space);
22                     printf("%-20.20s", s);
23                 }
24                 cout << endl;
25             }
26         }
27     }
28     else
29     {
30         cout << "query_sql_failed!" << endl;
31     }

```

运行结束后的控制台输出如图7所示。



```
Microsoft Visual Studio 调试控制台
=====result of 4 (1)=====
employeeNo      employeeName      salary
E2005001        喻自强           5800.00
E2008005        张小梅           5000.00
E2005004        张露             4100.00
E2006001        陈辉             4000.00
E2008004        李虹冰           3400.00
E2008001        陈诗杰           3200.00
E2008003        黄梅莹           3100.00
E2008002        张良             2700.00
E2005003        张小娟           2600.00
E2006002        韩梅             2600.00
E2006003        刘风             2500.00
E2007001        吴浮萍           2500.00
E2005002        张小梅           2400.00
E2007002        高代鹏           2000.00
E2005005        张小东           1800.00
=====
```

图 7: 4-(1)运行结果

任务4-(2): (C++)

```
1 char order_4_2 [] = "insert_into_customer(customerNo, _customerName, _address
, _telephone, _zip) _values_('C20080002', _'*****', _'***', _
'010-5422685', _'220501')";
2     printf(fence_begin, 4, 2);
3     res = mysql_query(&myConnect, order_4_2);
4     if (!res)
5     {
6         cout << "query_sql_succeeded!" << endl;;
7     }
8     else
9     {
10        cout << "query_sql_failed!" << endl;
11    }
```

上面代码块中中文内容未能正常显示. 变量order_4_2的值为"insert into customer(customerNo, customerName, address, telephone, zip) values ('C20080002', '泰康股份有限公司', '天津市', '010-5422685', '220501')".

运行结束后的控制台输出以及在mySql中查询客户ID为'C20080002'的结果如图8所示.

任务4-(3): (C++)

```
1 char order_4_3 [5][128] = { "SET_FOREIGN_KEY_CHECKS=_0",
2                             "SET_SQL_SAFE_UPDATES=_0",
3                             "delete_from_employee_where_salary >= 5000",
4                             "SET_SQL_SAFE_UPDATES=_1;",
5                             "SET_FOREIGN_KEY_CHECKS=_1" };
6     for (int i = 0; i < 5; i++) {
7         res = mysql_query(&myConnect, order_4_3[i]);
```

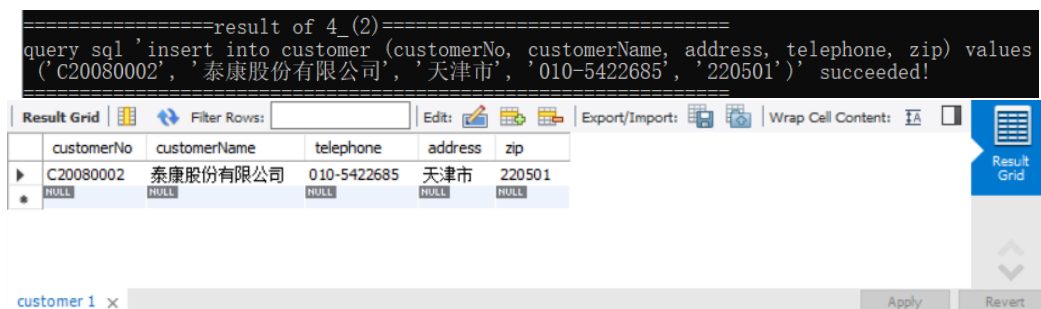


图 8: 4-(2)运行结果

```

8         if (!res)
9         {
10             cout << "query_sql_" << order_4_3[i] << "_succeeded!" <<
                endl;;
11         }
12         else
13         {
14             cout << "query_sql_" << order_4_3[i] << "_failed!" <<
                endl;
15             break;
16         }
17     }

```

运行结束后的控制台输出以及在删除语句执行前后mySql中查询薪水高于5000的员工信息如图9所示.

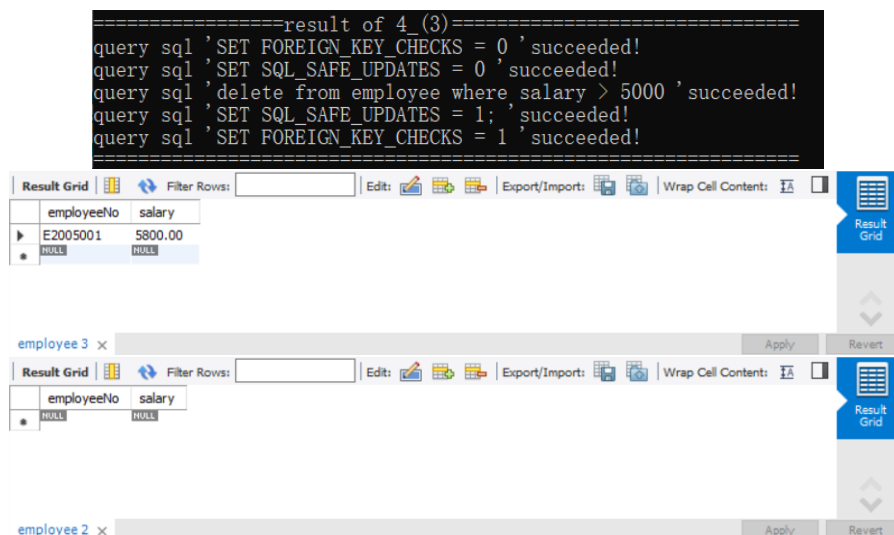


图 9: 4-(3)运行结果

任务4-(4): (C++)

```

1 char order_4_3[3][128] = { "SET_SQL_SAFE_UPDATES_=0" ,

```



```

2         "update_product_set_productPrice =
           productPrice * 0.5 where productPrice >
           1000",
3         "SET SQL_SAFE_UPDATES = 1" };
4     for (int i = 0; i < 3; i++) {
5         res = mysql_query(&myConnect, order_4_3[i]);
6         if (!res)
7         {
8             cout << "query_sql_" << order_4_3[i] << "_succeeded!" <<
              endl;;
9         }
10        else
11        {
12            cout << "query_sql_" << order_4_3[i] << "_failed!" <<
              endl;
13            break;
14        }
15    }

```

运行结束后的控制台输出以及在更新语句执行前后mySql中商品价格高于1000的商品信息如图10所示。

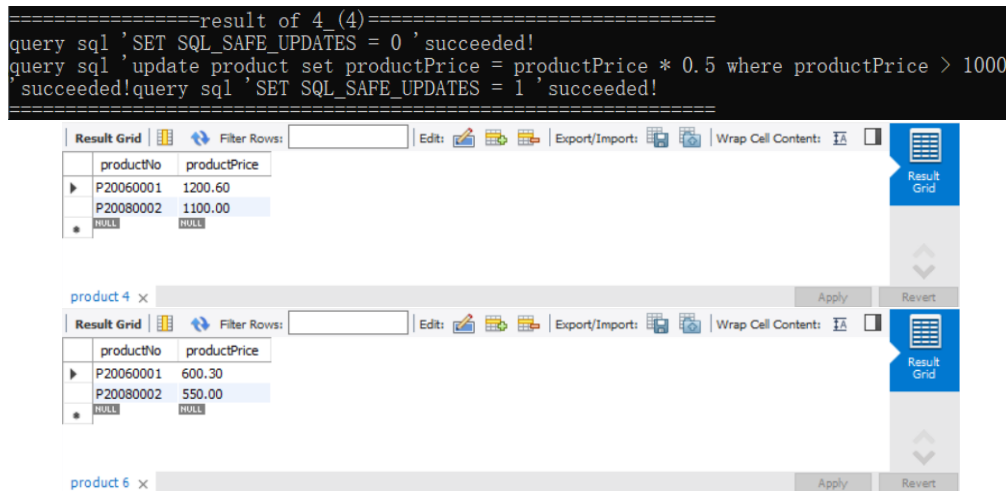


图 10: 4-(4)运行结果

任务5-(1): (C++)

```

1 char order_5_1[6][128] = { "SET SQL_SAFE_UPDATES = 0",
2 "set @sql = \"update employee set salary = salary + 200 where department =
   \"",
3 "PREPARE stmt FROM @sql",
4 "EXECUTE stmt",
5 "deallocate prepare stmt",
6 "SET SQL_SAFE_UPDATES = 1" };
7 char department[128];

```

```

8      cout << "Please input department Name:";
9      cin >> department;
10     strcat(department, " ");
11     strcat(order_5_1[1], department);
12     for (int i = 0; i < 6; i++) {
13         res = mysql_query(&myConnect, order_5_1[i]);
14         if (!res)
15         {
16             cout << "query_sql_" << order_5_1[i] << "_succeeded!" <<
17                 endl;;
18         }
19         else
20         {
21             cout << "query_sql_" << order_5_1[i] << "_failed!" <<
22                 endl;
23             break;
24         }
25     }

```

运行结束后的控制台输出以及在更新语句执行前后mySql中业务科员工工资信息如图11所示。

Please input department Name:业务科

```

=====result of 5_(1)=====
query sql 'SET SQL_SAFE_UPDATES = 0 'succeeded!
query sql 'set @sql = "update employee set salary = salary + 200 where department = '业务科'" 'succeeded!
query sql 'PREPARE stmt FROM @sql 'succeeded!
query sql 'EXECUTE stmt 'succeeded!
query sql 'deallocate prepare stmt 'succeeded!
query sql 'SET SQL_SAFE_UPDATES = 1 'succeeded!
=====

```

employeeNo	salary
E2005002	2400.00
E2005003	2600.00
E2005004	4100.00
E2005005	1800.00
E2006002	2600.00
E2006003	2500.00
E2007001	2500.00
E2008002	2700.00
E2008003	3100.00
E2008004	3400.00
NULL	NULL

employee 7 employee 8 × Apply Revert

employeeNo	salary
E2005002	2600.00
E2005003	2800.00
E2005004	4300.00
E2005005	2000.00
E2006002	2800.00
E2006003	2700.00
E2007001	2700.00
E2008002	2900.00
E2008003	3300.00
E2008004	3600.00
NULL	NULL

employee 9 employee 10 × Apply Revert

图 11: 5-(1)运行结果

任务5-(2): (C++)

```
1  char order_5_2[1][128] = { "select _customerName, _address, _telephone _from _  
   customer" };  
2      for (int i = 0; i < 1; i++) {  
3          res = mysql_query(&myConnect, order_5_2[i]);  
4          if (!res)  
5              {  
6                  result = mysql_store_result(&myConnect);  
7                  if (result)  
8                      {  
9                          int num_fields = mysql_num_fields(result);  
10                         MYSQL_FIELD* field;  
11                         char space[] = "____________________";  
12                         while (field = mysql_fetch_field(result)) {  
13                             char s[128];  
14                             sprintf(s, "%s%s", field->name, space);  
15                             printf("%-30.30s", s);  
16                         }  
17                         cout << endl;  
18                         while (sql_row = mysql_fetch_row(result))  
19                         {  
20                             for (int i = 0; i < num_fields; i++) {  
21                                 char s[128];  
22                                 sprintf(s, "%s%s", sql_row[i], space);  
23                                 printf("%-30.30s", s);  
24                             }  
25                             cout << endl;  
26                         }  
27                     }  
28                 }  
29             }  
30         else  
31         {  
32             cout << "query_sql '" << order_5_2[i] << "'_failed!" <<  
33                 endl;  
34             cout << mysql_error(&myConnect);  
35             break;  
36         }  
37     }
```

运行结束后的控制台输出如图12所示。

三、实验中遇到的困难及解决办法

配置C++ MySql时寻找教程¹花费了一定的时间。

```

=====result of 5_(2)=====
customerName      address      telephone
统一股份有限公司 天津市      022-3566021
兴隆股份有限公司 天津市      022-3562452
上海生物研究室   北京市      010-2121000
五一商厦          上海市      021-4532187
大地商城          北京市      010-1165152
联合股份有限公司 上海市      021-4568451
南昌市电脑研制中心 南昌市      0791-4412152
世界技术开发公司 上海市      021-4564512
万事达股份有限公司 天津市      022-4533141
红度股份有限公司 北京市      010-5421585
=====

```

图 12: 5-(2)运行结果

四、参考文献及致谢

1. <https://www.cnblogs.com/justinzhang/archive/2011/09/23/2185963.html>