

《数据库概论》实验二 高级SQL 实验报告

姓名：姜宁 学号：191840116 Email: 191840116@smail.nju.edu.cn

实验环境

Window10+MySQL8.0.26.0+Python3.9

实验过程

注：此部分解释基本的思路，具体实现中的一些细节写在实验遇到的困难及解决办法部分

Q1-1

```
delimiter //
drop procedure if exists product_info //
create procedure product_info(in productName varchar(40))
begin
    select om.customerNo,c.customerName,om.orderNo,od.quantity,
           od.quantity*od.price as totalPrice
    from OrderMaster om,OrderDetail od,Customer c,Product p
    where om.orderNo=od.orderNo and om.customerNo=c.customerNo
          and od.productNo=p.productNo and p.productName=productName
    order by (od.quantity*od.price) desc;
end //
delimiter ;
call product_info("32M DRAM");
```

创建过程 product_info,订货金额为 `od.quantity*od.price`,根据输入的参数 productName 进行 select并按订货金额降序输出即可。

用 `call product_info("32M DRAM")` 语句调用过程，结果如下：

	customerNo	customerName	orderNo	quantity	totalPrice
▶	C20050001	统一股份有限公司	200801090001	5	2500.00
	C20070002	世界技术开发公司	200803010001	8	1200.00
	C20080001	红度股份有限公司	200801090003	5	650.00
	C20050004	五一商厦	200803020001	2	200.00

Q1-2

```

delimiter //
drop procedure if exists earlier_employee_info //
create procedure earlier_employee_info(in employeeNo char(8))
begin
    select e1.employeeNo,e1.employeeName,e1.gender,e1.hireDate,e1.department
    from Employee e1,(
        select e2.hireDate,e2.department
        from Employee e2
        where e2.employeeNo=employeeNo) as newtable
    where e1.hireDate<newtable.hireDate and e1.department=newtable.department;
end //
delimiter ;
call earlier_employee_info("E2008005");

```

首先选出 newtable 表用来记录了 employeeNo 等于输入参数的员工的雇佣日期和部门，再用 Employee 表与 newtable 比较筛选即可。

用 call earlier_employee_info("E2008005") 语句调用过程，结果如下：

	employeeNo	employeeName	gender	hireDate	department
▶	E2005001	喻自强	M	1990-02-06 00:00:00	财务科

Q2-1

```

delimiter //
drop function if exists average_product //
create function average_product(productName varchar(40))
returns numeric(7,2) DETERMINISTIC
begin
    declare average numeric(7,2);
    select sum(newtable._sum)/sum(newtable._count) into average
    from (
        select p.productName as productName,
            od.quantity*od.price as _sum,od.quantity as _count
        from Product p,OrderDetail od
        where p.productName=productName and p.productNo=od.productNo
        ) as newtable;
    return average;
end //
delimiter ;
select distinct p.productName as productName,
    average_product(p.productName) as average
from Product p;

```

商品的订购平均价等于商品订购的总金额除以商品订购总量，所以需要首先构造 newtable 表包括商品名称，某订单商品订购金额，订购数量，再计算 sum(newtable._sum)/sum(newtable._count) 即可。

在select中调用函数 average_product，结果如下：

	productName	average
▶	32M DRAM	227.50
	17寸显示器	425.00
	120GB硬盘	215.00
	3.5寸软驱	350.00
	键盘	375.00
	VGA显示卡	337.50
	网卡	258.57
	Pentium 100CPU	290.00
	1G DDR	315.47
	52倍速光驱	362.50
	计算机字典	286.36
	9600bits/s调制...	342.00
	Pentium主板	245.71
	硕泰克SL-K8...	265.71
	龙基777FT纯...	215.33

Q2-2

```

delimiter //
drop function if exists quantity_product //
create function quantity_product(productNo char(9))
    returns int DETERMINISTIC
begin
    declare res int;
    select sum(od.quantity) into res
    from OrderDetail od
    where od.productNo=productNo;
    return res;
end //
delimiter ;
select p.productNo as productNo, p.productName as productName,
       quantity_product(p.productNo) as quantity
from Product p
where quantity_product(p.productNo)>4;

```

从 OrderDetail 表中统计订购数量即可。

在select中调用函数 quantity_product，结果如下：

	productNo	productName	quantity
▶	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	Pentium100CPU	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

Q3-1

```

delimiter //
drop trigger if exists check_price //
create trigger check_price before insert on Product
for each row
begin
    if new.productPrice>1000 then set new.productPrice=1000;
    end if ;
end //
delimiter ;

```

根据题意，触发器应当在插入前触发，所以是 `before insert`。

在orderDB文件中运行后使用下面的代码测试：

```

select p.productNo as productNo,p.productName as productName,
       p.productPrice as productPrice
from Product p;

```

结果如下：

	productNo	productName	productPrice
▶	P20050001	32M DRAM	80.70
	P20050002	17寸显示器	700.00
	P20050003	120GB硬盘	300.00
	P20050004	3.5寸软驱	35.00
	P20050005	键盘	100.60
	P20060001	VGA显示卡	1000.00
	P20060002	网卡	66.00
	P20060003	Pentium100CPU	200.00
	P20070001	1G DDR	256.00
	P20070002	52倍速光驱	200.00
	P20070003	计算机字典	100.00
	P20070004	9600bits/s调制...	320.00
	P20080001	Pentium主板	890.00
	P20080002	硕泰克SL-K8...	1000.00
	P20080003	龙基777FT纯...	900.00

Q3-2

```

delimiter //
drop trigger if exists add_salary //
create trigger add_salary after insert on OrderMaster
for each row
begin
    update Employee e set e.salary=e.salary*1.05
    where e.hireDate>='19920101' and e.employeeNo=new.employeeNo;
    update Employee e set e.salary=e.salary*1.08
    where e.hireDate<'19920101' and e.employeeNo=new.employeeNo;
end //
delimiter ;

```

根据题意，触发器应当在插入新订单之后触发，所以是 `after insert`，根据不同条件更新相应的薪水即可。

在orderDB文件中运行后使用下面的代码测试：

```

select e.employeeNo,e.employeeName,e.salary
from Employee e;

```

结果如下：

	employeeNo	employeeName	salary
▶	E2005001	喻自强	5800.00
	E2005002	张小梅	3023.31
	E2005003	张小娟	2866.50
	E2005004	张露	4100.00
	E2005005	张小东	1800.00
	E2006001	陈辉	4000.00
	E2006002	韩梅	2600.00
	E2006003	刘风	2500.00
	E2007001	吴浮萍	2500.00
	E2007002	高代鹏	2000.00
	E2008001	陈诗杰	3456.00
	E2008002	张良	2976.75
	E2008003	黄梅莹	3348.00
	E2008004	李虹冰	3672.00
	E2008005	张小梅	5000.00

Q4-1

```

from pymysql import connect
import pymysql.cursors

connection=pymysql.connect(host='localhost',port=3306,user='root',password='xxxx
xxxxxxxx',
db='OrderDB',charset='utf8mb4',cursorclass=pymysql.cursors.DictCursor)

try:
    cursor=connection.cursor()
    sql=''
    select new_table.employeeNo as employeeNo,
        new_table.employeeName as employeeName,
        new_table.salary as salary,
        new_table.rownum as rownum
    from (
        select ee.employeeNo,ee.employeeName,ee.salary,
            @rownum:=@rownum+1 as num_temp,
            @incnum:=case
            when @nowsal=ee.salary then @incnum
            when @nowsal:=ee.salary then @rownum
            end as rownum
        from(
            select e.employeeNo,e.employeeName,e.salary
            from Employee e
            order by e.salary desc
        ) as ee,
        (
            select @rownum:=0,@nowsal:=NULL,@incnum:=0
        ) r
    ) as new_table
    where new_table.rownum<=20;
    '''
    cursor.execute(sql)
    result=cursor.fetchall()

```

```

for data in result:
    print(data)
except Exception:print("error!")

cursor.close()
connection.close()

```

成功连接到 orderDB 表后通过游标运行sql语句，下面解释一下sql语句：为了打印薪水排名，定义了三个变量：@rownum 是一个自加变量，@incnum 是实际排名，nowsal 表示上一个比较完的薪水值；首先将 Employee 表按薪水降序排序,代码块


```

@rownum:=@rownum+1 as num_temp,
@incnum:=case
    when @nowsal=ee.salary then @incnum -- 相等那么排名不变
    when @nowsal:=ee.salary then @rownum -- 否则等于一直自增的rownum并且更新nowsal
end as rownum

```

用来处理薪水相同的排名，正确结果应当为排名相等并且不影响后面的排名，例如salary为2500的两人排名都为11，紧跟其后的salary为2400的人排名应当是13而不是12，所以用两个变量处理即可实现。

查询结果如下：

 Windows PowerShell

```

PS C:\Users\ASUS\Desktop\数据库\exp2> python .\Q4_1.py
{'employeeNo': 'E2005001', 'employeeName': '喻自强', 'salary': Decimal('5800.00'), 'rownum': '1'}
{'employeeNo': 'E2008005', 'employeeName': '张小梅', 'salary': Decimal('5000.00'), 'rownum': '2'}
{'employeeNo': 'E2005004', 'employeeName': '张露', 'salary': Decimal('4100.00'), 'rownum': '3'}
{'employeeNo': 'E2006001', 'employeeName': '陈辉', 'salary': Decimal('4000.00'), 'rownum': '4'}
{'employeeNo': 'E2008004', 'employeeName': '李虹冰', 'salary': Decimal('3400.00'), 'rownum': '5'}
{'employeeNo': 'E2008001', 'employeeName': '陈诗杰', 'salary': Decimal('3200.00'), 'rownum': '6'}
{'employeeNo': 'E2008003', 'employeeName': '黄梅莹', 'salary': Decimal('3100.00'), 'rownum': '7'}
{'employeeNo': 'E2008002', 'employeeName': '张良', 'salary': Decimal('2700.00'), 'rownum': '8'}
{'employeeNo': 'E2005003', 'employeeName': '张小娟', 'salary': Decimal('2600.00'), 'rownum': '9'}
{'employeeNo': 'E2006002', 'employeeName': '韩梅', 'salary': Decimal('2600.00'), 'rownum': '9'}
{'employeeNo': 'E2006003', 'employeeName': '刘凤', 'salary': Decimal('2500.00'), 'rownum': '11'}
{'employeeNo': 'E2007001', 'employeeName': '吴浮萍', 'salary': Decimal('2500.00'), 'rownum': '11'}
{'employeeNo': 'E2005002', 'employeeName': '张小梅', 'salary': Decimal('2400.00'), 'rownum': '13'}
{'employeeNo': 'E2007002', 'employeeName': '高代鹏', 'salary': Decimal('2000.00'), 'rownum': '14'}
{'employeeNo': 'E2005005', 'employeeName': '张小东', 'salary': Decimal('1800.00'), 'rownum': '15'}
PS C:\Users\ASUS\Desktop\数据库\exp2>

```

Q4-2

```

from pymysql import connect
import pymysql.cursors

connection=pymysql.connect(host='localhost',port=3306,user='root',password='xxxx
XXXXXXXXXX',
db='OrderDB',charset='utf8mb4',cursorclass=pymysql.cursors.DictCursor)

try:
    cursor=connection.cursor()
    sql=''
    insert Customer values(
        'c20080002','泰康股份有限公司', '010-5422685', '天津市', '220501'
    );
    ...
    cursor.execute(sql)
    sql=''
    select * from Customer;
    ...

```

```

cursor.execute(sql)
result=cursor.fetchall()
for data in result:
    print(data)
except Exception:print("error!")

cursor.close()
connection.close()

```

成功连接 orderDB 后通过游标运行sql语句即可。（Q4-3, Q4-4, Q4-5同理）结果如下：

Windows PowerShell

```

PS C:\Users\ASUS\Desktop\数据库\exp2> python .\Q4_2.py
{'customerNo': 'C20050001', 'customerName': '统一股份有限公司', 'telephone': '022-3566021', 'address': '天津市', 'zip': '220012'}
{'customerNo': 'C20050002', 'customerName': '兴隆股份有限公司', 'telephone': '022-3562452', 'address': '天津市', 'zip': '220301'}
{'customerNo': 'C20050003', 'customerName': '上海生物研究室', 'telephone': '010-2121000', 'address': '北京市', 'zip': '108001'}
{'customerNo': 'C20050004', 'customerName': '五一商厦', 'telephone': '021-4532187', 'address': '上海市', 'zip': '210100'}
{'customerNo': 'C20060001', 'customerName': '大地商城', 'telephone': '010-1165152', 'address': '北京市', 'zip': '100803'}
{'customerNo': 'C20060002', 'customerName': '联合股份有限公司', 'telephone': '021-4568451', 'address': '上海市', 'zip': '210100'}
{'customerNo': 'C20070001', 'customerName': '南昌市电脑研制中心', 'telephone': '0791-4412152', 'address': '南昌市', 'zip': '330046'}
{'customerNo': 'C20070002', 'customerName': '世界技术开发公司', 'telephone': '021-4564512', 'address': '上海市', 'zip': '210230'}
{'customerNo': 'C20070003', 'customerName': '万事达股份有限公司', 'telephone': '022-4533141', 'address': '天津市', 'zip': '220400'}
{'customerNo': 'C20080001', 'customerName': '红度股份有限公司', 'telephone': '010-5421585', 'address': '北京市', 'zip': '100800'}
{'customerNo': 'C20080002', 'customerName': '泰康股份有限公司', 'telephone': '010-5422685', 'address': '天津市', 'zip': '220501'}
PS C:\Users\ASUS\Desktop\数据库\exp2>

```

Q4-3

```

from pymysql import connect
import pymysql.cursors

connection=pymysql.connect(host='localhost',port=3306,user='root',password='xxxx
XXXXXXXXXX',
db='OrderDB',charset='utf8mb4',cursorclass=pymysql.cursors.DictCursor)

try:
    cursor=connection.cursor()

    sql=''
    delete from Employee e
    where e.salary>5000;
    ''

    cursor.execute(sql)
    sql=''
    select e.employeeNo,e.employeeName,e.salary
    from Employee e;
    ''

    cursor.execute(sql)
    result=cursor.fetchall()
    for data in result:
        print(data)
except Exception:print("error!")

cursor.close()
connection.close()

```

结果如下：


```
PS C:\Users\ASUS\Desktop\数据库\exp2> python .\Q4_3.py
{'employeeNo': 'E2005002', 'employeeName': '张小梅', 'salary': Decimal('2400.00')}
{'employeeNo': 'E2005003', 'employeeName': '张小娟', 'salary': Decimal('2600.00')}
{'employeeNo': 'E2005004', 'employeeName': '张露', 'salary': Decimal('4100.00')}
{'employeeNo': 'E2005005', 'employeeName': '张小东', 'salary': Decimal('1800.00')}
{'employeeNo': 'E2006001', 'employeeName': '陈辉', 'salary': Decimal('4000.00')}
{'employeeNo': 'E2006002', 'employeeName': '韩梅', 'salary': Decimal('2600.00')}
{'employeeNo': 'E2006003', 'employeeName': '刘风', 'salary': Decimal('2500.00')}
{'employeeNo': 'E2007001', 'employeeName': '吴浮萍', 'salary': Decimal('2500.00')}
{'employeeNo': 'E2007002', 'employeeName': '高代鹏', 'salary': Decimal('2000.00')}
{'employeeNo': 'E2008001', 'employeeName': '陈诗杰', 'salary': Decimal('3200.00')}
{'employeeNo': 'E2008002', 'employeeName': '张良', 'salary': Decimal('2700.00')}
{'employeeNo': 'E2008003', 'employeeName': '黄梅莹', 'salary': Decimal('3100.00')}
{'employeeNo': 'E2008004', 'employeeName': '李虹冰', 'salary': Decimal('3400.00')}
{'employeeNo': 'E2008005', 'employeeName': '张小梅', 'salary': Decimal('5000.00')}
PS C:\Users\ASUS\Desktop\数据库\exp2>
```

Q4-4

```
from pymysql import connect
import pymysql.cursors

connection=pymysql.connect(host='localhost',port=3306,user='root',password='xxxx
xxxxxxxx',
db='OrderDB',charset='utf8mb4',cursorclass=pymysql.cursors.DictCursor)

try:
    cursor=connection.cursor()

    sql=''
    update Product p
    set p.productPrice=p.productPrice*0.5
    where p.productPrice>1000;
    ''

    cursor.execute(sql)
    sql=''
    select p.productNo,p.productName,p.productPrice
    from Product p;
    ''

    cursor.execute(sql)
    result=cursor.fetchall()
    for data in result:
        print(data)
except Exception:print("error!")

cursor.close()
connection.close()
```

结果如下:

```
PS C:\Users\ASUS\Desktop\数据库\exp2> python .\Q4_4.py
{'productNo': 'P20050001', 'productName': '32M DRAM', 'productPrice': Decimal('80.70')}
{'productNo': 'P20050002', 'productName': '17寸显示器', 'productPrice': Decimal('700.00')}
{'productNo': 'P20050003', 'productName': '120GB硬盘', 'productPrice': Decimal('300.00')}
{'productNo': 'P20050004', 'productName': '3.5寸软驱', 'productPrice': Decimal('35.00')}
{'productNo': 'P20050005', 'productName': '键盘', 'productPrice': Decimal('100.60')}
{'productNo': 'P20060001', 'productName': 'VGA显示卡', 'productPrice': Decimal('600.30')}
{'productNo': 'P20060002', 'productName': '网卡', 'productPrice': Decimal('66.00')}
{'productNo': 'P20060003', 'productName': 'Pentium100CPU', 'productPrice': Decimal('200.00')}
{'productNo': 'P20070001', 'productName': '1G DDR', 'productPrice': Decimal('256.00')}
{'productNo': 'P20070002', 'productName': '52倍速光驱', 'productPrice': Decimal('200.00')}
{'productNo': 'P20070003', 'productName': '计算机字典', 'productPrice': Decimal('100.00')}
{'productNo': 'P20070004', 'productName': '9600bits/s调制解调', 'productPrice': Decimal('320.00')}
{'productNo': 'P20080001', 'productName': 'Pentium主板', 'productPrice': Decimal('890.00')}
{'productNo': 'P20080002', 'productName': '硕泰克SL-K8AN-RL主板', 'productPrice': Decimal('550.00')}
{'productNo': 'P20080003', 'productName': '龙基777FT纯平显示器', 'productPrice': Decimal('900.00')}
PS C:\Users\ASUS\Desktop\数据库\exp2>
```

Q5-1

```
from pymysql import connect
import pymysql.cursors

connection=pymysql.connect(host='localhost',port=3306,user='root',password='XXXX
XXXXXXXXXX',
db='OrderDB',charset='utf8mb4',cursorclass=pymysql.cursors.DictCursor)

try:
    cursor=connection.cursor()
    department="业务科"
    sql=''
    update Employee e
    set e.salary=e.salary+200
    where e.department=%s;      -- 占位符
    '''

    cursor.execute(sql,department)
    sql=''
    select e.employeeNo,e.employeeName,e.department,e.salary
    from Employee e;
    '''

    cursor.execute(sql)
    result=cursor.fetchall()
    for data in result:
        print(data)
except Exception:print("error!")

cursor.close()
connection.close()
```

通过占位符方式将部门变为外部输入参数，结果如下：

```
Windows PowerShell
PS C:\Users\ASUS\Desktop\数据库\exp2> python .\Q5_1.py
({'employeeNo': 'E2005001', 'employeeName': '喻自强', 'department': '财务科', 'salary': Decimal('5800.00')},
 {'employeeNo': 'E2005002', 'employeeName': '张小梅', 'department': '业务科', 'salary': Decimal('2600.00')},
 {'employeeNo': 'E2005003', 'employeeName': '张小娟', 'department': '业务科', 'salary': Decimal('2800.00')},
 {'employeeNo': 'E2005004', 'employeeName': '张露', 'department': '业务科', 'salary': Decimal('4300.00')},
 {'employeeNo': 'E2005005', 'employeeName': '张小东', 'department': '业务科', 'salary': Decimal('2000.00')},
 {'employeeNo': 'E2006001', 'employeeName': '陈辉', 'department': '办公室', 'salary': Decimal('4000.00')},
 {'employeeNo': 'E2006002', 'employeeName': '韩梅', 'department': '业务科', 'salary': Decimal('2800.00')},
 {'employeeNo': 'E2006003', 'employeeName': '刘风', 'department': '业务科', 'salary': Decimal('2700.00')},
 {'employeeNo': 'E2007001', 'employeeName': '吴浮萍', 'department': '业务科', 'salary': Decimal('2700.00')},
 {'employeeNo': 'E2007002', 'employeeName': '高代鹏', 'department': '办公室', 'salary': Decimal('2000.00')},
 {'employeeNo': 'E2008001', 'employeeName': '陈诗杰', 'department': '财务科', 'salary': Decimal('3200.00')},
 {'employeeNo': 'E2008002', 'employeeName': '张良', 'department': '业务科', 'salary': Decimal('2900.00')},
 {'employeeNo': 'E2008003', 'employeeName': '黄梅莹', 'department': '业务科', 'salary': Decimal('3300.00')},
 {'employeeNo': 'E2008004', 'employeeName': '李虹冰', 'department': '业务科', 'salary': Decimal('3600.00')},
 {'employeeNo': 'E2008005', 'employeeName': '张小梅', 'department': '财务科', 'salary': Decimal('5000.00')})
PS C:\Users\ASUS\Desktop\数据库\exp2>
```

Q5-2

```
from pymysql import connect
import pymysql.cursors

connection=pymysql.connect(host='localhost',port=3306,user='root',password='XXXX
XXXXXXXXXX',
db='OrderDB',charset='utf8mb4',cursorclass=pymysql.cursors.DictCursor)

try:
    cursor=connection.cursor()
    sql=''
    select c.customerName,c.address,c.telephone
    from Customer c;
    '''
    cursor.execute(sql)
    result=cursor.fetchone()
    tplt="{0:{3}<10}\t{1:{3}<4}\t{2:{3}<12}"
    print(tplt.format("客户名称","客户地址","客户电话",chr(12288)))
    print(tplt.format(result['customerName'],result['address'],
    result['telephone'],chr(12288)))
    while result:
        print(tplt.format(result['customerName'],result['address'],
        result['telephone'],chr(12288)))
        result=cursor.fetchone()
except Exception:print("error!")

cursor.close()
connection.close()
```

通过 `cursor.fetchone` 方法获取结果的第一条数据，再用while循环获取所有数据，这样做的好处相比于 `fetchall` 是开销较小，为了打印美观这里用了python的 `format` 格式化输出。结果如下：

```
Windows PowerShell
PS C:\Users\ASUS\Desktop\数据库\exp2> python .\Q5_2.py
客户名称      客户地址      客户电话
统一股份有限公司      天津市      022-3566021
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
PS C:\Users\ASUS\Desktop\数据库\exp2>
```

实验中遇到的困难及解决办法

1.结束符问题

mysql默认的结束符是`;`，所以过程和函数中写完select语句加上分号之后会遇到这个问题：`MySQL: statement is incomplete, expecting ;` 这是因为mysql遇到`;`之后就会默认语句块结束，但是此时并没有写到end，所以会报错语句不完整。改正方法是使用 `delimiter //` 语句将结束符改为 `//`，这样便不会报错了。

2.python连接mysql及使用

python调用mysql需要使用 `pymysql` 库，首先需要通过 `pymysql` 的 `connect()` 方法声明一个mysql连接对象，连接成功后，再调用 `cursor()` 方法获得mysql的操作游标，利用游标来执行SQL语句。

参考文献及致谢

触发器的使用参考了[周伯通的麦田](#)的trigger实例详解；

python连接MySQL参考了[崔庆才](#)的MySQL存储。