

《数据库概论》实验二：高级 SQL 实验报告

张涵之 191220154 191220154@smail.nju.edu.cn

实验环境

操作系统：Windows 10, 64 位操作系统, 基于 x64 的处理器

软件版本：MySQL Community 8.0.26.0, PyCharm Community Edition 2021.2.3

实验过程

1. 声明并调用存储过程，完成以下任务：

- 1) 存储过程功能：根据输入商品名称，找到订购该商品的客户编号、客户名称、订单编号、订货数量和订货金额，并按订货金额降序输出；调用该存储过程查询“32M DRAM”商品。

设置了 $quantity * price$ ，由于 OrderMaster 中的 orderSum 未正确同步。

```
• drop procedure if exists find_order;
• create procedure find_order(in product_name varchar(40))
    select Customer.customerNo as customerNo, Customer.customerName as customerName,
        FindOrder2.orderNo as orderNo, FindOrder2.quantity as quantity, FindOrder2.OrderSum as orderSum
    from Customer, (
        select OrderMaster.customerNo as CustomerNo, FindOrder.orderNo as orderNo,
            FindOrder.quantity as quantity, FindOrder.OrderSum as OrderSum
        from OrderMaster, (
            select OrderDetail.orderNo as orderNo, OrderDetail.quantity as quantity,
                OrderDetail.quantity * OrderDetail.price as OrderSum
            from OrderDetail, (
                select Product.productNo as productNo
                from Product
                where Product.productName = product_name) as FindProduct
            where OrderDetail.productNo = FindProduct.productNo) as FindOrder
        where OrderMaster.orderNo = FindOrder.OrderNo) as FindOrder2
    where Customer.customerNo = FindOrder2.CustomerNo
    order by orderSum desc;

• call find_order("32M DRAM");
```

	customerNo	customerName	orderNo	quantity	orderSum
▶	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

- 2) 存储过程功能：根据输入的员工编号，查询比该员工雇佣日期早的同部门的员工编号、姓名、性别、雇佣日期、所属部门；调用该存储过程查询比员工编号“E2008005”雇佣日期早的同一部门的员工编号、姓名、性别、雇佣日期、所属部门。

- `drop procedure if exists find_predecessor;`
- `create procedure find_predecessor(in employee_no char(8))`
`select employeeNo, employeeName, gender, hireDate, department`
`from Employee, (`
`select hireDate as findDate, department as findDepartment`
`from Employee`
`where employeeNo = employee_no) as findEmployee`
`where department = findDepartment and hireDate < findDate;`
- `call find_predecessor("E2008005");`

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

2. 声明并调用存储函数，完成以下任务：

- 1) 存储函数功能：根据输入的商品名称，返回该商品订购平均价；

调用该存储函数查询商品基本信息表中所有商品名称及其订购平均价。

采用加权平均，返回 $\text{totalPrice} / \text{totalQuantity}$ （统一保留两位小数）。

- `drop function if exists get_avg_price;`
`DELIMITER //`
- `create function get_avg_price(product_name varchar(40))`
`returns numeric(7,2)`
`begin`
`declare totalQuantity numeric(7,2);`
`declare totalPrice numeric(7,2);`
`select sum(quantity), sum(price * quantity) into totalQuantity, totalPrice`
`from OrderDetail`
`where productNo in (`
`select productNo`
`from Product`
`where productName = product_name);`
`return round(totalPrice / totalQuantity, 2);`
`end //`
`DELIMITER ;`
- `select productName, get_avg_price(productName) as avgPrice from Product;`

	productName	avgPrice
▶	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
	Pentium100CPU	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

- 2) 存储函数功能：根据输入的商品编号，统计该商品的销售总量；
调用该存储函数查询销售总量大于4的商品编号、商品名称及销售数量。

- ```
drop function if exists get_total_quantity;
DELIMITER //
```
- ```
create function get_total_quantity(product_no char(9))  
returns int  
begin  
    declare totalQuantity int;  
    select sum(quantity) into totalQuantity  
    from OrderDetail  
    where productNo = product_no;  
    return totalQuantity;  
end //
```

```
DELIMITER ;
```
- ```
select productNo, productName, get_total_quantity(productNo) as totalQuantity
from Product
where get_total_quantity(productNo) > 4;
```

|   | productNo | productName     | totalQuantity |
|---|-----------|-----------------|---------------|
| ▶ | 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             |

### 3. 创建触发器，完成以下任务：

- 1) 当插入一项商品时，如果商品价格大于1000，则将其设置为1000。

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

```
DELIMITER ;
```
- ```
insert Product values('P20090007','什么玩意','图书',800.00);
```
- ```
insert Product values('P20100008','不会起名','设备',1200.00);
```
- ```
select * from Product;
```

	P20090007	什么玩意	图书	800.00
	P20100008	不会起名	设备	1000.00

2) 员工完成订单时，薪水增加5%；如果员工1992年前入职再增加3%。

- `drop trigger if exists update_salary;`
`DELIMITER //`
- `create trigger update_salary before insert on OrderMaster for each row`

```

begin
    declare newHireDate datetime;
    select hireDate into newHireDate from Employee where Employee.employeeNo = new.employeeNo;
    if newHireDate < '19920101'
        then update Employee set salary = salary * 1.08 where Employee.employeeNo = new.employeeNo;
        else update Employee set salary = salary * 1.05 where Employee.employeeNo = new.employeeNo;
    end if;
end //
DELIMITER ;

```
- `select employeeNo, employeeName, hireDate, salary from Employee`
`where employeeNo = 'E2005001' or employeeNo = 'E2005003';`
- `insert OrderMaster values('202112240007','C20050001','E2005001','20211224',0.00,'I000000020');`
- `insert OrderMaster values('202112250001','C20050001','E2005003','20211225',0.00,'I000000030');`
- `select employeeNo, employeeName, hireDate, salary from Employee`
`where employeeNo = 'E2005001' or employeeNo = 'E2005003';`

	employeeNo	employeeName	hireDate	salary	salary
▶	E2005001	喻自强	1990-02-06 00:00:00	5800.00	6264.00
	E2005003	张小娟	1992-03-28 00:00:00	2600.00	2730.00

4. 使用高级程序设计语言访问SQL并执行如下的查询、插入、删除、更新：

1) 查询职工工资按从高到低排序的前20的职工编号、职工姓名和工资；

使用python语言在PyCharm中链接MySQL数据库。

```

import pymysql
db = pymysql.connect(host="127.0.0.1", port=3306, user="root", password=
                        database="OrderDB", charset="utf8")
cursor = db.cursor()
sql_4_1 = """
    SELECT employeeNo, employeeName, salary FROM Employee
    ORDER BY salary DESC LIMIT 20;
"""

```

```

('E2005001', '喻自强', Decimal('5800.00'))
('E2008005', '张小梅', Decimal('5000.00'))
('E2005004', '张露', Decimal('4100.00'))
('E2006001', '陈辉', Decimal('4000.00'))
('E2008004', '李虹冰', Decimal('3400.00'))
('E2008001', '陈诗杰', Decimal('3200.00'))
('E2008003', '黄梅莹', Decimal('3100.00'))
('E2008002', '张良', Decimal('2700.00'))
('E2005003', '张小娟', Decimal('2600.00'))
('E2006002', '韩梅', Decimal('2600.00'))
('E2006003', '刘凤', Decimal('2500.00'))
('E2007001', '吴浮萍', Decimal('2500.00'))
('E2005002', '张小梅', Decimal('2400.00'))
('E2007002', '高代鹏', Decimal('2000.00'))
('E2005005', '张小东', Decimal('1800.00'))

```


- 2) 为客户表插入一条新的客户信息，编号“C20080002”，名称“泰康股份有限公司”，电话“010-5422685”，地址“天津市”，邮编“220501”；

```
sql_4_2 = """
    INSERT Customer VALUES('C20080002', '泰康股份有限公司',
    '010-5422686', '天津市', '220501');
"""

sql_4_2_helper = """
    SELECT * FROM Customer;
"""

if __name__ == '__main__':
    cursor.execute(sql_4_2)
    cursor.execute(sql_4_2_helper)
    results = cursor.fetchall()
    for row in results:
        print(row)
    db.close()
```

```
('C20080002', '泰康股份有限公司', '010-5422686', '天津市', '220501')
```

为方便表示，此处仅截图刚刚插入的这一行数据。

- 3) 删除员工表中薪水高于5000的员工信息；

```
sql_4_3 = """
    DELETE FROM Employee WHERE salary > 5000;
"""

sql_4_3_helper = """
    SELECT * FROM Employee;
"""

if __name__ == '__main__':
    cursor.execute(sql_4_3)
```

```
(1451, 'Cannot delete or update a parent row: a foreign key constraint fails
```

执行这一步时出现了报错，而我感到这似乎是符合设计原数据库中外键设计逻辑的，假如直接删除这些员工，他们卖出的订单怎么办呢？

```
Error Code: 1175. You are using safe update mode and you tried to update a table ...
```

直接在MySQL Workbench中运行删除代码也出现类似报错。

- 4) 更新商品基本信息表中价格超过1000的商品价格变为原来的50%；

```
sql_4_4 = """
    UPDATE Product SET productPrice = productPrice * 0.5
    WHERE productPrice > 1000;
"""

sql_4_4_helper = """
    SELECT * from Product;
"""
```

可以看到修改商品价格不涉及外键，因此顺利地完成了。

```
( 'P20050001', '32M DRAM', '内存', Decimal('80.70'))
( 'P20050002', '17寸显示器', '显示器', Decimal('700.00'))
( 'P20050003', '120GB硬盘', '存储器', Decimal('300.00'))
( 'P20050004', '3.5寸软驱', '设备', Decimal('35.00'))
( 'P20050005', '键盘', '设备', Decimal('100.60'))
( 'P20060001', 'VGA显示卡', '显示器', Decimal('600.30'))
( 'P20060002', '网卡', '设备', Decimal('66.00'))
( 'P20060003', 'Pentium100CPU', '处理器', Decimal('200.00'))
( 'P20070001', '1G DDR', '内存', Decimal('256.00'))
( 'P20070002', '52倍速光驱', '设备', Decimal('200.00'))
( 'P20070003', '计算机字典', '图书', Decimal('100.00'))
( 'P20070004', '9600bits/s调制解调', '设备', Decimal('320.00'))
( 'P20080001', 'Pentium主板', '主板', Decimal('890.00'))
( 'P20080002', '硕泰克SL-K8AN-RL主板', '主板', Decimal('550.00'))
( 'P20080003', '龙基777FT纯平显示器', '显示器', Decimal('900.00'))
```

5. 使用高级程序设计语言中的动态SQL功能完成如下的任务：

- 1) 为“业务科”（作为外部输入参数）所有员工增加200的薪水。

```
sql_5_1 = """
    UPDATE Employee SET salary = salary + 200
    WHERE department = %s;
"""

sql_5_1_helper = """
    SELECT employeeNo, employeeName, department, salary
    FROM Employee WHERE department = '业务科';
"""

if __name__ == '__main__':
    cursor.execute(sql_5_1, '业务科')
    cursor.execute(sql_5_1_helper)
```

```
( 'E2005002', '张小梅', '业务科', Decimal('2600.00'))
( 'E2005003', '张小娟', '业务科', Decimal('2930.00'))
( 'E2005004', '张露', '业务科', Decimal('4300.00'))
( 'E2005005', '张小东', '业务科', Decimal('2000.00'))
( 'E2006002', '韩梅', '业务科', Decimal('2800.00'))
( 'E2006003', '刘凤', '业务科', Decimal('2700.00'))
( 'E2007001', '吴浮萍', '业务科', Decimal('2700.00'))
( 'E2008002', '张良', '业务科', Decimal('2900.00'))
( 'E2008003', '黄梅莹', '业务科', Decimal('3300.00'))
( 'E2008004', '李虹冰', '业务科', Decimal('3600.00'))
```

这里“张小娟”的薪水之前在测试3-2（员工完成订单时薪水增加）修改过没有改回去，因此这名员工的薪水可能和您的预期输出不同。

- 2) 查询客户表中的客户名称、客户地址及客户电话并输出。

```
sql_5_2 = """
    SELECT customerName, address, telephone FROM Customer;
"""
```

```
('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')
```

实验中遇到的困难及解决办法

1. 存储过程和函数的定义与使用：阅读网上的介绍和示例，仿照着写。
2. PyCharm环境配置：出现报错时再搜索报错信息，如my.ini中的默认时区设置会导致PyCharm连接数据库时报错time zone显示乱码。
3. 其他报错（如4-3中删除时出现外键约束报错）：结合课程理论部分内容思考数据库的原理和原则，以理解方法为主，不强求一定要输出结果。

参考文献及致谢

课程课件（主要是“存储过程与函数简介.pptx”）

部分高级语言的使用参考了网络，如runnoob.com的Python和MySQL数据库连接部分：<https://www.runoob.com/python3/python3-mysql.html>