

南京信息工程大学数据库系统原理实验（实习）报告

实验（实习）名称_____连接查询_____实验（实习）日期 2024-4-21 得分_____指导教师_____
系 计算机学院 专业 计科

一、实验目的

1. 掌握连接查询中基本连接条件和筛选条件的设计方法
2. 掌握多表连接的条件表示方法和筛选条件的设计方法
3. 掌握 INNER JOIN 内连接的设计方法
4. 掌握 OUTER JOIN 外连接的设计方法
5. 掌握连接查询中进行分组、分组后筛选、结果排序的方法

二、实验内容

在 GoodsOrder 数据库中，用 SQL 语句实现如下查询：

- 27、查询订购了商品、所在省市为“江苏南京”的客户编号、客户姓名及其全部订单信息。
- 28、查询订购了商品、所在省市为“江苏南京”的客户编号、客户姓名及其订购“食品”类订单信息。
- 29、查询在“2020-2-17”以后订购了“食品”类商品的客户编号、客户姓名及其订单信息。
- 30、查询订购了单价在 50 元（含）及以上、并且订购数量在 2 个（含）及以上的客户编号、客户姓名及其订购商品信息。
- 31、查询订购了“新新文化用品制造厂”生产的商品、且单个订单的订购数量超过不少于 2 的客户编号、客户姓名及其相应订单信息。
- 32、求出每笔订单的金额。
- 33、查询订购了同一编号的商品超过一次的客户编号、及其相应商品编号。
- 34、查询订购了同一编号的商品超过一次的客户编号、及其相应订单信息。
- 35、查询订购了同一编号的商品超过一次的客户编号、客户姓名及其相应商品编号。
- 36、查询至少有连续两天购买下单的客户编号。
- 37、查询和“张小林”同一个省市的其他客户情况。
- 38、查询客户信息及其订购商品的编号，若客户未订购任何商品，则商品编号为 NULL。
- 39、查询所有商品信息及其订购的客户编号，若商品未被订购过，则客户编号为 NULL。
- 40、统计每个客户的订单数量，若客户未订购任何商品，则其订单数为 0。
- 41、查询 35 岁及以上的客户编号、客户姓名及其订单信息，若客户未订购任何商品，也要输出其编号和姓名。
- 42、查询订购了商品的客户的编号、姓名、及其每个订单的金额，并按订单金额由高到低排序。
- 43、统计订购了商品的女客户的客户编号及其订单数量，不包含未订购商品的客户。
- 44、统计各个女客户的客户编号及其订单数量，若客户未订购任何商品，则其订单数为 0，并按订单数量由高到低排序。
- 45、若客户下了 2 个及以上的食品类订单，请把客户编号及其食品类订单个

数统计出来。

46、统计每个客户所下订单的总金额，未订购的客户不做统计，按总金额由大到小排序输出。

47、统计客户编号及其所有订单的总金额，若客户未订购任何商品，则总金额为 0。按总金额由大到小排序输出。

48、统计客户所下订单的总金额，未订购的客户不做统计，按总金额由大到小排序输出前 3 个客户的编号及其总金额。

三、实验过程与结果

(1) StudentCourse 实验

1、查询每个学生的情况和其选修课程的课程号及成绩。

- 设计的 SQL 语句：

```
SELECT Student.*, 课程号, 成绩
FROM Student , StuCourse
WHERE Student.学号 = StuCourse.学号
```

- 语句执行结果如下：

	学号	姓名	专业名	性别	出生时间	总学分	备注	课程号	成绩
1	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好生	1001	98
2	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好生	1002	95
3	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好生	2001	88
4	070102	王红	计算机科学与技术	女	1988-12-20 00:00:00	80	NULL	1001	97
5	070102	王红	计算机科学与技术	女	1988-12-20 00:00:00	80	NULL	1002	75
6	070102	王红	计算机科学与技术	女	1988-12-20 00:00:00	80	NULL	2001	82
7	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目	1001	78
8	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目	1002	55
9	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目	2001	81
10	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL	1001	82
11	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL	1002	80
12	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL	3001	85
13	070202	王波	电子信息工程	男	1989-02-18 00:00:00	76	多次获...	1001	96
14	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL	1002	95
15	070202	王波	电子信息工程	男	1989-02-18 00:00:00	76	多次获...	3001	98
16	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL	1001	72
17	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL	1002	70
18	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL	3001	80
19	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL	2001	98

2、查找计算机科学与技术专业学生的情况和其选修课程的课程号及成绩。

- 设计的 SQL 语句：

```
SELECT Student.*, 课程号, 成绩
FROM Student , StuCourse
WHERE Student.学号 = StuCourse.学号 AND 专业名='计算机科学与技术'
```

- 语句执行结果如下：

结果		消息							
	学号	姓名	专业名	性别	出生时间	总学分	备注	课程号	成绩
1	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好生	1001	98
2	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好生	1002	95
3	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好生	2001	88
4	070102	王红	计算机科学与技术	女	1988-12-20 00:00:00	80	NULL	1001	97
5	070102	王红	计算机科学与技术	女	1988-12-20 00:00:00	80	NULL	1002	75
6	070102	王红	计算机科学与技术	女	1988-12-20 00:00:00	80	NULL	2001	82
7	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目	1001	78
8	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目	1002	55
9	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目	2001	81

3、查找成绩低于 60 分的学生情况和不及格课程的课程号及成绩。

- 设计的 SQL 语句:

```
SELECT Student.*, 课程号, 成绩
```

```
FROM Student , StuCourse
```

```
WHERE Student.学号 = StuCourse.学号 AND 成绩<60
```

- 语句执行结果如下:

结果		消息							
	学号	姓名	专业名	性别	出生时间	总学分	备注	课程号	成绩
1	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目	1002	55

4、查询每个学生的情况和其选修课程的课程名及成绩。其中输出的成绩用等级代替: ≥ 90 为优, ≥ 80 且 < 90 为良, ≥ 70 且 < 80 为中, ≥ 60 且 < 70 为及格, < 60 为不及格。。

- 设计的 SQL 语句:

```
SELECT a.*, 课程名, 成绩等级=
```

```
CASE
```

```
WHEN 成绩 $\geq$ 90 THEN '优'
```

```
WHEN 成绩 $\geq$ 80 AND 成绩 $<$  90 THEN '良'
```

```
WHEN 成绩 $\geq$ 70 AND 成绩 $<$  80 THEN '中'
```

```
WHEN 成绩 $\geq$ 60 AND 成绩 $<$  70 THEN '及格'
```

```
ELSE '不及格'
```

```
END
```

```
FROM Student a, StuCourse b, Course c
```

```
WHERE a.学号 = b.学号 AND b.课程号= c.课程号
```

- 语句执行结果如下:

	学号	姓名	专业名	性别	出生时间	总学分	备注	课程名	成绩等级
1	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好学生	高等数学1	优
2	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好学生	高等数学2	优
3	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好学生	程序设计基础	良
4	070102	王红	计算机科学与技术	女	1988-12-20 00:00:00	80	NULL	高等数学1	优
5	070102	王红	计算机科学与技术	女	1988-12-20 00:00:00	80	NULL	高等数学2	中
6	070102	王红	计算机科学与技术	女	1988-12-20 00:00:00	80	NULL	程序设计基础	良
7	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目	高等数学1	中
8	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目	高等数学2	不及格
9	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目	程序设计基础	良
10	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL	高等数学1	良
11	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL	高等数学2	良
12	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL	电路基础	良
13	070202	王波	电子信息工程	男	1989-02-18 00:00:00	76	多次获奖学金	高等数学1	优
14	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL	高等数学2	优
15	070202	王波	电子信息工程	男	1989-02-18 00:00:00	76	多次获奖学金	电路基础	优
16	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL	高等数学1	中
17	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL	高等数学2	中
18	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL	电路基础	良
19	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL	程序设计基础	优

5、查询所有学生及其选课情况，若学生未选课程，也要将该生的信息输出。

- 设计的 SQL 语句：

```
SELECT a.*, 课程号, 成绩
FROM Student a LEFT JOIN StuCourse b
ON a.学号=b.学号
```

- 语句执行结果如下：

	学号	姓名	专业名	性别	出生时间	总学分	备注	课程号	成绩
1	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好学生	1001	98
2	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好学生	1002	95
3	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好学生	2001	88
4	070102	王红	计算机科学与技术	女	1988-12-20 00:00:00	80	NULL	1001	97
5	070102	王红	计算机科学与技术	女	1988-12-20 00:00:00	80	NULL	1002	75
6	070102	王红	计算机科学与技术	女	1988-12-20 00:00:00	80	NULL	2001	82
7	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目	1001	78
8	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目	1002	55
9	070105	朱江	计算机科学与技术	男	1990-01-10 00:00:00	78	有补考科目	2001	81
10	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL	1001	82
11	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL	1002	80
12	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL	3001	85
13	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL	1002	95
14	070202	王波	电子信息工程	男	1989-02-18 00:00:00	76	多次获奖学金	1001	96
15	070202	王波	电子信息工程	男	1989-02-18 00:00:00	76	多次获奖学金	3001	98
16	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL	1001	72
17	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL	1002	70
18	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL	3001	80
19	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL	2001	98
20	070207	朱平平	电子信息工程	女	1990-01-10 00:00:00	74	NULL	NULL	NULL
21	070208	李进	电子信息工程	男	1989-09-12 00:00:00	74	NULL	NULL	NULL

(2) GoodsOrder 实验

27、查询订购了商品、所在省市为“江苏南京”的客户编号、客户姓名及其全部订单信息。

- 分析：查询结果信息来自 CustomerInfo、Orderlist，故为多表查询；目标列为客户编号、客户姓名及全部订单信息，WHERE 使用连接谓词“=”进行查询。

- 设计的 SQL 语句如下：

```
SELECT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
```

```
FROM CustomerInfo, OrderList
```

```
WHERE 所在省市 = '江苏南京' AND CustomerInfo.客户编号 = OrderList.客户编号
```

- 语句执行结果如下：

SQLQuery1.sql - (L... (WZY\15532 (51))*)

```
SELECT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList
WHERE 所在省市 = '江苏南京' AND CustomerInfo.客户编号 = OrderList.客户编号
```

结果 消息

	客户编号	客户姓名	客户编号	商品编号	订购时间	数量	需要日期	付款方式	送货方式
1	100001	张小林	100001	10010001	2020-02-18 12:20:00.000	2	2020-02-20 00:00:00.000	支付宝	客户自提
2	100001	张小林	100001	30010001	2020-02-10 12:30:00.000	10	2020-02-20 00:00:00.000	网银转账	送货上门
3	100006	王芳芳	100006	10020001	2020-02-23 09:00:00.000	5	2020-02-26 00:00:00.000	信用卡	送货上门

- 进一步的思考与练习：

(1) 另一种方法：

```
SELECT A.客户编号, A.客户姓名, B.*
```

```
FROM CustomerInfo A JOIN OrderList B ON A.客户编号 = B.客户编号
```

```
WHERE A.所在省市 = '江苏南京'
```

(2) 查询订购了商品、所在省市为“江苏”的客户编号、客户姓名及其全部订单信息，如下：

```
SELECT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
```

```
FROM CustomerInfo, OrderList
```

```
WHERE 所在省市 LIKE '江苏%' AND CustomerInfo.客户编号 = OrderList.客户编号
```

SQLQuery1.sql - (L... (WZY\15532 (51))*)

```
SELECT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList
WHERE 所在省市 LIKE '江苏%' AND CustomerInfo.客户编号 = OrderList.客户编号
```

结果 消息

	客户编号	客户姓名	客户编号	商品编号	订购时间	数量	需要日期	付款方式	送货方式
1	100001	张小林	100001	10010001	2020-02-18 12:20:00.000	2	2020-02-20 00:00:00.000	支付宝	客户自提
2	100001	张小林	100001	30010001	2020-02-10 12:30:00.000	10	2020-02-20 00:00:00.000	网银转账	送货上门
3	100002	李红红	100002	10010001	2020-02-18 13:00:00.000	1	2020-02-21 00:00:00.000	微信支付	客户自提
4	100002	李红红	100002	50020001	2020-02-18 13:20:00.000	1	2020-02-21 00:00:00.000	微信支付	客户自提
5	100006	王芳芳	100006	10020001	2020-02-23 09:00:00.000	5	2020-02-26 00:00:00.000	信用卡	送货上门

28、查询订购了商品、所在省市为“江苏南京”的客户编号、客户姓名及其订购“食品”类订单信息。

- 分析：查询结果信息来自 CustomerInfo、Orderlist、GoodsInfo，故为多表查询；目标列为客户编号、客户姓名及订购商品为“食品”类的信息，WHERE 中进行条件判断。

- 设计的 SQL 语句:

```
SELECT DISTINCT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList, GoodsInfo
WHERE 所在省市 = '江苏南京' AND CustomerInfo.客户编号 = OrderList.客
户编号 AND OrderList.商品编号 = GoodsInfo.商品编号 AND 商品类别 = '食品'
```

- 语句执行结果如下:



SQLQuery1.sql - (L... (WZY\15532 (51)))

```
SELECT DISTINCT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList, GoodsInfo
WHERE 所在省市 = '江苏南京' AND CustomerInfo.客户编号 = OrderList.客户编号 AND OrderList.商品编号 = GoodsInfo.商品编号 AND 商品类别 = '食品'
```

	客户编号	客户姓名	客户编号	商品编号	订购时间	数量	需要日期	付款方式	送货方式
1	100001	张小林	100001	10010001	2020-02-18 12:20:00.000	2	2020-02-20 00:00:00.000	支付宝	客户自提
2	100006	王芳芳	100006	10020001	2020-02-23 09:00:00.000	5	2020-02-26 00:00:00.000	信用卡	送货上门

- 进一步的思考与练习:

(1) 使用 JOIN 方法:

```
SELECT A.客户编号, A.客户姓名, B.*
FROM CustomerInfo A
JOIN OrderList B ON A.客户编号 = B.客户编号
JOIN GoodsInfo C ON C.商品编号 = B.商品编号
WHERE A.所在省市 = '江苏南京' AND C.商品类别 = '食品'
```

(2) 查询订购了商品、所在省市为“江苏南京”的客户编号、客户姓名及其订购“文具”类订单信息, 如下:

```
SELECT DISTINCT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList, GoodsInfo
WHERE 所在省市 = '江苏南京' AND CustomerInfo.客户编号 = OrderList.客
户编号 AND OrderList.商品编号 = GoodsInfo.商品编号 AND 商品类别 = '文具'
```



SQLQuery1.sql - (L... (WZY\15532 (51)))

```
SELECT DISTINCT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList, GoodsInfo
WHERE 所在省市 = '江苏南京' AND CustomerInfo.客户编号 = OrderList.客户编号 AND OrderList.商品编号 = GoodsInfo.商品编号 AND 商品类别 = '文具'
```

	客户编号	客户姓名	客户编号	商品编号	订购时间	数量	需要日期	付款方式	送货方式
1	100001	张小林	100001	30010001	2020-02-10 12:30:00.000	10	2020-02-20 00:00:00.000	网银转账	送货上门

29、查询在“2020-2-17”以后订购了“食品”类商品的客户编号、客户姓名及其订单信息。

- 分析: 查询结果信息来自 CustomerInfo、Orderlist, GoodsInfo, 故为多表查询; 目标列为客户编号、客户姓名及订购商品为“食品”类的信息, WHERE 使用连接谓词“=”进行查询。

- 设计的 SQL 语句:

```
SELECT DISTINCT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList, GoodsInfo
WHERE 订购时间 >= '2020-2-17' AND CustomerInfo.客户编号 = OrderList.客
户编号 AND OrderList.商品编号 = GoodsInfo.商品编号 AND 商品类别 = '食品'
```

- 语句执行结果如下:

SQLQuery0.sql - (L...er (WZY\15532 (53)))

SQLQuery1.sql - (L... (WZY\15532 (51)))

```
SELECT DISTINCT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList, GoodsInfo
WHERE 订购时间 >= '2020-2-17' AND CustomerInfo.客户编号 = OrderList.客户编号 AND OrderList.商品编号 = GoodsInfo.商品编号 AND 商品类别 = '食品'
```

结果 消息

	客户编号	客户姓名	客户编号	商品编号	订购时间	数量	需要日期	付款方式	送货方式
1	100001	张小林	100001	10010001	2020-02-18 12:20:00.000	2	2020-02-20 00:00:00.000	支付宝	客户自提
2	100002	李红红	100002	10010001	2020-02-18 13:00:00.000	1	2020-02-21 00:00:00.000	微信支付	客户自提
3	100006	王芳芳	100006	10020001	2020-02-23 09:00:00.000	5	2020-02-26 00:00:00.000	信用卡	送货上门

- 进一步的思考与练习：

- (1) 使用 JOIN 方法：

```
SELECT A.客户编号,A.客户姓名,B.*
FROM CustomerInfo A
JOIN OrderList B ON A.客户编号 = B.客户编号
JOIN GoodsInfo C ON C.商品编号 = B.商品编号
WHERE 订购时间 >= '2020-2-17' AND C.商品类别 = '食品'
```

30、查询订购了单价在 50 元（含）及以上、并且订购数量在 2 个（含）及以上的客户编号、客户姓名及其订购商品信息。

- 分析：查询结果信息来自 CustomerInfo、Orderlist, GoodsInfo，故为多表查询；目标列为客户编号、客户姓名及订购商品为“食品”类的信息，WHERE 使用连接谓词进行查询。

- 设计的 SQL 语句：

```
SELECT DISTINCT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList, GoodsInfo
WHERE 数量 >= 2 AND CustomerInfo.客户编号 = OrderList.客户编号 AND
OrderList.商品编号 = GoodsInfo.商品编号 AND 单价 >= 50
```

- 语句执行结果如下：

SQLQuery0.sql - (L...er (WZY\15532 (53)))

SQLQuery1.sql - (L... (WZY\15532 (51)))

```
SELECT DISTINCT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList, GoodsInfo
WHERE 数量 >= 2 AND CustomerInfo.客户编号 = OrderList.客户编号 AND OrderList.商品编号 = GoodsInfo.商品编号 AND 单价 >= 50
```

结果

消息

	客户编号	客户姓名	客户编号	商品编号	订购时间	数量	需要日期	付款方式	送货方式
1	100001	张小林	100001	10010001	2020-02-18 12:20:00.000	2	2020-02-20 00:00:00.000	支付宝	客户自提
2	100004	赵明	100004	50020002	2020-02-19 10:40:00.000	2	2020-02-28 00:00:00.000	信用卡	送货上门

- 进一步的思考与练习：

- (1) 使用 JOIN 方法：

```
SELECT A.客户编号,A.客户姓名,B.*
FROM CustomerInfo A
JOIN OrderList B ON A.客户编号 = B.客户编号
JOIN GoodsInfo C ON C.商品编号 = B.商品编号
WHERE B.数量 >= 2 AND C.单价 >= 50
```

31、查询订购了“新新文化用品制造厂”生产的商品、且单个订单的订购数量超过不少于 2 的客户编号、客户姓名及其相应订单信息。

- 分析：查询结果信息来自 CustomerInfo、Orderlist, GoodsInfo，故为多表查询；目标列为客户编号、客户姓名及相应订单信息，WHERE 使用连接谓词“=”和“>=”进行查询。

- 设计的 SQL 语句:

```
SELECT DISTINCT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList, GoodsInfo
WHERE 数量 >= 2 AND CustomerInfo.客户编号 = OrderList.客户编号 AND
OrderList.商品编号 = GoodsInfo.商品编号 AND 生产商 = '新新文化用品制造厂'
```

- 语句执行结果如下:



SQLQuery1.sql - (L... (WZY\15532 (51)))

```
SELECT DISTINCT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList, GoodsInfo
WHERE 数量 >= 2 AND CustomerInfo.客户编号 = OrderList.客户编号 AND OrderList.商品编号 = GoodsInfo.商品编号 AND 生产商 = '新新文化用品制造厂'
```

	客户编号	客户姓名	客户编号	商品编号	订购时间	数量	需要日期	付款方式	送货方式
1	100001	张小林	100001	30010001	2020-02-10 12:30:00.000	10	2020-02-20 00:00:00.000	网银转账	送货上门
2	100004	赵明	100004	30010002	2020-02-19 11:00:00.000	10	2020-02-28 00:00:00.000	信用卡	送货上门


- 进一步的思考与练习:

- (1) 使用 JOIN 方法:

```
SELECT A.客户编号, A.客户姓名, B.*
FROM CustomerInfo A
JOIN OrderList B ON A.客户编号 = B.客户编号
JOIN GoodsInfo C ON C.商品编号 = B.商品编号
WHERE B.数量 >= 2 AND C.生产商 = '新新文化用品制造厂'
```

- (2) 查询订购了“新新文化用品制造厂”或“宇一饮料公司”生产的商品、且单个订单的订购数量超过不少于 2 的客户编号、客户姓名及其相应订单信息, 如下:

```
SELECT DISTINCT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList, GoodsInfo
WHERE 数量 >= 2 AND CustomerInfo.客户编号 = OrderList.客户编号 AND
OrderList.商品编号 = GoodsInfo.商品编号 AND (生产商 = '新新文化用品制造厂'
OR 生产商 = '宇一饮料公司')
```



SQLQuery1.sql - (L... (WZY\15532 (51)))

```
SELECT DISTINCT CustomerInfo.客户编号, CustomerInfo.客户姓名, OrderList.*
FROM CustomerInfo, OrderList, GoodsInfo
WHERE 数量 >= 2 AND CustomerInfo.客户编号 = OrderList.客户编号 AND OrderList.商品编号 = GoodsInfo.商品编号 AND (生产商 = '新新文化用品制造厂' OR 生产商 = '宇一饮料公司')
```

	客户编号	客户姓名	客户编号	商品编号	订购时间	数量	需要日期	付款方式	送货方式
1	100001	张小林	100001	10010001	2020-02-18 12:20:00.000	2	2020-02-20 00:00:00.000	支付宝	客户自提
2	100001	张小林	100001	30010001	2020-02-10 12:30:00.000	10	2020-02-20 00:00:00.000	网银转账	送货上门
3	100004	赵明	100004	30010002	2020-02-19 11:00:00.000	10	2020-02-28 00:00:00.000	信用卡	送货上门

32、求出每笔订单的金额。

- 分析: 查询结果信息来自 CustomerInfo、Orderlist, GoodsInfo, 故为多表查询; 目标列为客户编号、商品编号、订购时间和订单金额, WHERE 使用连接谓词“=”进行查询。

- 设计的 SQL 语句:


```
SELECT DISTINCT b.客户编号, b.商品编号, b.订购时间, (b.数量 * c.单价) AS '订单金额'
FROM CustomerInfo a, OrderList b, GoodsInfo c
WHERE a.客户编号 = b.客户编号 AND b.商品编号 = c.商品编号
```

- 语句执行结果如下:

SQLQuery0.sql - (l...er (WZY\15532 (53)))

SQLQuery1.sql - (l... (WZY\15532 (51)))

```
SELECT DISTINCT b.客户编号,b.商品编号,b.订购时间,(b.数量 * c.单价) AS '订单金额'
FROM CustomerInfo a,OrderList b ,GoodsInfo c
WHERE a.客户编号 = b.客户编号 AND b.商品编号 = c.商品编号
```

 结果  消息

	客户编号	商品编号	订购时间	订单金额
1	100001	10010001	2020-02-18 12:20:00.000	100
2	100001	30010001	2020-02-10 12:30:00.000	35
3	100002	10010001	2020-02-18 13:00:00.000	50
4	100002	50020001	2020-02-18 13:20:00.000	30
5	100004	20180002	2020-02-19 10:00:00.000	120
6	100004	30010002	2020-02-19 11:00:00.000	56
7	100004	50020002	2020-02-19 10:40:00.000	160
8	100005	40010001	2020-02-20 08:00:00.000	76
9	100005	40010002	2020-02-20 08:20:00.000	60
10	100006	10020001	2020-02-23 09:00:00.000	175

- 进一步的思考与练习:

- (1) 使用 JOIN 方法:

```
SELECT B.客户编号,A.客户姓名,B.订购时间,(B.数量*C.单价) AS '订单金额'
FROM CustomerInfo A
JOIN OrderList B ON A.客户编号 = B.客户编号
JOIN GoodsInfo C ON C.商品编号 = B.商品编号
```

- (2) 求出订单金额大于等于 100 的订单信息, 如下:

```
SELECT DISTINCT b.客户编号,b.商品编号,b.订购时间,(b.数量 * c.单价) AS '订单金额'
FROM CustomerInfo a,OrderList b,GoodsInfo c
WHERE a.客户编号 = b.客户编号 AND b.商品编号 = c.商品编号 AND (b.数量 * c.单价 >= 100)
```

SQLQuery0.sql - (l...er (WZY\15532 (53)))

SQLQuery1.sql - (l... (WZY\15532 (51)))

```
SELECT DISTINCT b.客户编号,b.商品编号,b.订购时间,(b.数量 * c.单价) AS '订单金额'
FROM CustomerInfo a,OrderList b,GoodsInfo c
WHERE a.客户编号 = b.客户编号 AND b.商品编号 = c.商品编号 AND (b.数量 * c.单价 >= 100)
```

结果

消息

	客户编号	商品编号	订购时间	订单金额
1	100001	10010001	2020-02-18 12:20:00.000	100
2	100004	20180002	2020-02-19 10:00:00.000	120
3	100004	50020002	2020-02-19 10:40:00.000	160
4	100006	10020001	2020-02-23 09:00:00.000	175

33、查询订购了同一编号的商品超过一次的客户编号、及其相应商品编号。

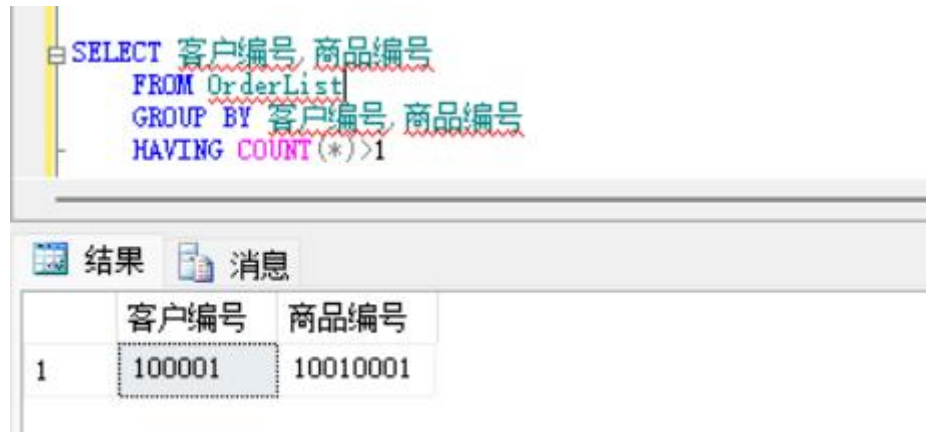
- 分析: 查询结果信息来自 Orderlist, 故为单表查询; 目标列为客户编号、订购编号, 使

用 GROUP BY 对其进行分组，选择数量大于等于 2 的目标。

- 设计的 SQL 语句：

```
SELECT 客户编号,商品编号
FROM OrderList
GROUP BY 客户编号,商品编号
HAVING COUNT(*)>1
```

- 语句执行结果如下：



	客户编号	商品编号
1	100001	10010001

34、查询订购了同一编号的商品超过一次的客户编号、及其相应订单信息。

- 分析：查询结果信息来自 Orderlist，故为单表查询；目标列为客户编号及相应订单信息，使用多重查询。

- 设计的 SQL 语句：

```
SELECT *
FROM OrderList
WHERE 客户编号 IN (
    SELECT 客户编号
    FROM OrderList
    GROUP BY 客户编号,商品编号
    HAVING COUNT(*)>1
) AND 商品编号 IN (
    SELECT 商品编号
    FROM OrderList
    GROUP BY 客户编号,商品编号
    HAVING COUNT(*)>1
)
```

- 语句执行结果如下：



	客户编号	商品编号	订购时间	数量	需要日期	付款方式	送货方式
1	100001	10010001	2020-02-18 12:20:00.000	2	2020-02-20 00:00:00.000	支付宝	客户自提
2	100001	10010001	2020-03-01 00:00:00.000	3	2020-03-05 00:00:00.000	支付宝	送货 上门
3	100001	10010001	2020-03-03 00:00:00.000	2	2020-03-07 00:00:00.000	支付宝	送货 上门

- 进一步的思考与练习：

(1) 另一种方法：

```

SELECT *
  FROM OrderList O1
 WHERE EXISTS (
    SELECT 1
    FROM OrderList O2
   WHERE O1.客户编号 = O2.客户编号
     AND O1.商品编号 = O2.商品编号
   GROUP BY O2.客户编号, O2.商品编号
  HAVING COUNT(*) > 1
 );

```

35、查询订购了同一编号的商品超过一次的客户编号、客户姓名及其相应商品编号。

- 分析：查询结果信息来自 Orderlist, CustomerInfo, 故为多表查询；目标列为客户编号、客户姓名及商品编号，使用多重查询。

- 设计的 SQL 语句：

```

SELECT DISTINCT CustomerInfo.客户编号,客户姓名,商品编号
  FROM OrderList, CustomerInfo
 WHERE CustomerInfo.客户编号 IN (
    SELECT 客户编号
      FROM OrderList
     GROUP BY 客户编号,商品编号
    HAVING COUNT(*)>1
 ) AND OrderList.商品编号 IN (
    SELECT 商品编号
      FROM OrderList
     GROUP BY 客户编号,商品编号
    HAVING COUNT(*)>1
 )

```

- 语句执行结果如下：

结果		消息	
	客户编号	客户姓名	商品编号
1	100001	张小林	10010001

36、查询至少有连续两天购买下单的客户编号。

- 分析：查询结果信息来自 Orderlist, 故为单表查询；目标列为客户编号，将 OrderList 客户编号相同的信息连接，在使用 DateDIFF 函数判断日期。

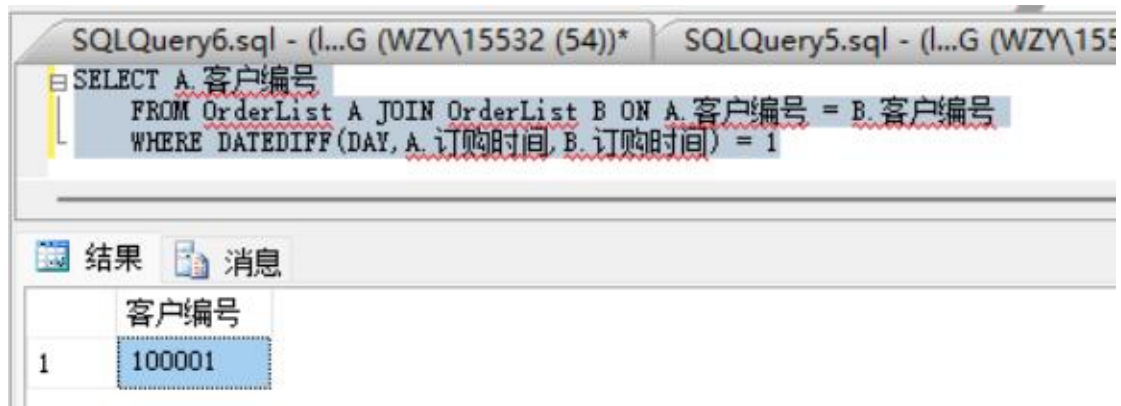
- 设计的 SQL 语句：

```

SELECT A.客户编号
  FROM OrderList A JOIN OrderList B ON A.客户编号 = B.客户编号
 WHERE DATEDIFF(DAY,A.订购时间,B.订购时间) = 1

```

- 语句执行结果如下：



37、查询和“张小林”同一个省市的其他客户情况。

- 分析：查询结果信息来自 CustomerInfo，故为单表查询；目标行为同一个省市其他客户情况，先将客户张小林的所在省市选择出来，再选择所在省市等于此并且姓名不等于张小林的用户，进行嵌套查询。

- 设计的 SQL 语句：

```
SELECT *
FROM CustomerInfo
WHERE 客户姓名!= '张小林' AND 所在省市 IN (
    SELECT 所在省市
    FROM CustomerInfo
    WHERE 客户姓名 = '张小林'
)
```

- 语句执行结果如下：

	客户编号	客户姓名	出生日期	性别	所在省市	联系电话	微信号	VIP	备注
1	100002	李红红	1991-03-22 00:00:00.000	女	江苏苏州	13908899120	13908899120	1	金牌客户
2	100003	王晓美	1986-08-20 00:00:00.000	女	上海市	02166552101	wxid_0021001	0	新客户
3	100006	王芳芳	1996-05-01 00:00:00.000	女	江苏南京	13709092011	wxid_7890921	0	NULL

- 进一步的思考与练习：

(1) 使用 JOIN 语句写，如下：

```
SELECT A.*
FROM CustomerInfo A JOIN CustomerInfo B ON A.所在省市 = B.所在省市
WHERE A.客户姓名 != '张小林' AND B.客户姓名 = '张小林'
```

38、查询客户信息及其订购商品的编号，若客户未订购任何商品，则商品编号为 NULL。

- 分析：查询结果信息来自 CustomerInfo 和 OrderList，故为多表查询；将这两个通过左外连接，即得到查询结果。

- 设计的 SQL 语句：

```
SELECT A.*,B.商品编号
FROM CustomerInfo A LEFT JOIN OrderList B ON A.客户编号 = B.客户编号
```

- 语句执行结果如下：

	客户编号	客户姓名	出生日期	性别	所在省市	联系电话	微信号	VIP	备注	商品编号
4	100002	李红红	1991-03-22 00:00:00.000	女	江苏苏州	13908899120	13908899120	1	金牌客户	50020001
5	100003	王晓美	1986-08-20 00:00:00.000	女	上海市	02166552101	wxid_0021001	0	新客户	NULL
6	100004	赵明	1992-03-28 00:00:00.000	男	河南郑州	13809900118	NULL	0	新客户	20180002
7	100004	赵明	1992-03-28 00:00:00.000	男	河南郑州	13809900118	NULL	0	新客户	30010002
8	100004	赵明	1992-03-28 00:00:00.000	男	河南郑州	13809900118	NULL	0	新客户	50020002
9	100005	张帆一	1990-08-10 00:00:00.000	男	山东烟台	13880933201	NULL	0	NULL	40010001
10	100005	张帆一	1990-08-10 00:00:00.000	男	山东烟台	13880933201	NULL	0	NULL	40010002
11	100006	王芳芳	1996-05-01 00:00:00.000	女	江苏南京	13709092011	wxid_7890921	0	NULL	10020001

39、查询所有商品信息及其订购的客户编号，若商品未被订购过，则客户编号为 NULL。

- 分析：与上一题类似，OrderList 右外连接 CustomerInfo。
- 设计的 SQL 语句：

SELECT B.*,A.客户编号

FROM OrderList A RIGHT JOIN GoodsInfo B ON A.商品编号 = B.商品编号

- 语句执行结果如下：

	商品编号	商品类别	商品名称	品牌	单价	生产商	保质期	库存量	备注	客户编号
1	10010001	食品	咖啡	宇一	50	宇一饮料公司	2021-08-31 00:00:00.000	100	NULL	100001
2	10010001	食品	咖啡	宇一	50	宇一饮料公司	2021-08-31 00:00:00.000	100	NULL	100002
3	10010002	食品	苹果汁	宇一	5.2	宇一饮料公司	2020-12-31 00:00:00.000	500	NULL	NULL
4	10020001	食品	大米	健康	35	健康粮食生产基地	2020-12-20 00:00:00.000	100	NULL	100006
5	10020002	食品	面粉	健康	18	健康粮食生产基地	2021-01-20 00:00:00.000	20	NULL	NULL
6	20180001	服装	运动服	天天	200	天天服饰公司	2000-01-01 00:00:00.000	5	有断码	NULL
7	20180002	服装	T恤	天天	120	天天服饰公司	2000-01-01 00:00:00.000	10	NULL	100004
8	30010001	文具	签字笔	新新	3.5	新新文化用品制造厂	2000-01-01 00:00:00.000	100	NULL	100001
9	30010002	文具	文件夹	新新	5.6	新新文化用品制造厂	2000-01-01 00:00:00.000	50	NULL	100004
10	40010001	图书	营养菜谱	新华	38	新华图书出版公司	2000-01-01 00:00:00.000	12	NULL	100005
11	40010002	图书	豆浆...	新华	20	新华图书出版公司	2000-01-01 00:00:00.000	20	NULL	100005
12	50020001	体育用品	羽毛球拍	美好	30	美好体育用品公司	2000-01-01 00:00:00.000	30	NULL	100002
13	50020002	体育用品	篮球	美好	80	美好体育用品公司	2000-01-01 00:00:00.000	20	NULL	100004
14	50020003	体育用品	足球	美好	65	美好体育用品公司	2000-01-01 00:00:00.000	20	NULL	NULL

40、统计每个客户的订单数量，若客户未订购任何商品，则其订单数为 0。

- 分析：与上一题类似思路，CustomerInfo 左外连接 OrderList。
- 设计的 SQL 语句：

SELECT A.客户编号,COUNT(B.商品编号)AS 订单数量

FROM CustomerInfo A LEFT JOIN OrderList B ON A.客户编号 = B.客户编号

GROUP BY A.客户编号

- 语句执行结果如下：

	客户编号	订单数量
1	100001	2
2	100002	2
3	100003	0
4	100004	3
5	100005	2
6	100006	1

41、查询 35 岁及以上的客户编号、客户姓名及其订单信息，若客户未订购任何商品，也要输出其编号和姓名。

- 分析：查询结果信息来自 CustomerInfo 和 OrderList，故为多表查询；CustomerInfo 左外连接 OrderList，再使用 DateDiff()函数算年龄。

- 设计的 SQL 语句：

```
SELECT A.客户编号,A.客户姓名,B.商品编号,B.订购时间,B.数量, B.需要日期,B.付款方式,B.送货方式
```

```
FROM CustomerInfo A LEFT JOIN OrderList B ON A.客户编号 = B.客户编号
```

```
WHERE DATEDIFF(YEAR,A.出生日期,GetDate())>=35
```

- 语句执行结果如下：

	客户编号	客户姓名	商品编号	订购时间	数量	需要日期	付款方式	送货方式
1	100001	张小林	10010001	2020-02-18 12:20:00.000	2	2020-02-20 00:00:00.000	支付宝	客户自提
2	100001	张小林	30010001	2020-02-10 12:30:00.000	10	2020-02-20 00:00:00.000	网银转账	送货上门
3	100003	王晓美	NULL	NULL	NULL	NULL	NULL	NULL
4	100005	张帆一	40010001	2020-02-20 08:00:00.000	2	2020-02-27 00:00:00.000	支付宝	送货上门
5	100005	张帆一	40010002	2020-02-20 08:20:00.000	3	2020-02-27 00:00:00.000	支付宝	送货上门

42、查询订购了商品的客户的编号、姓名、及其每个订单的金额，并按订单金额由高到低排序。

- 分析：查询结果信息来自 CustomerInfo、OrderList 和 GoodsInfo，故为多表查询；使用 JOIN 连接这三个表，再计算出每个订单金额。

- 设计的 SQL 语句：

```
SELECT A.客户编号,A.客户姓名,(B.数量* C.单价) AS '订单金额'
```

```
FROM CustomerInfo A JOIN OrderList B ON A.客户编号 = B.客户编号 JOIN
```

```
GoodsInfo C ON B.商品编号 = C.商品编号
```

```
ORDER BY B.数量* C.单价 DESC
```

- 语句执行结果如下：

	客户编号	客户姓名	订单金额
1	100006	王芳芳	175
2	100004	赵明	160
3	100004	赵明	120
4	100001	张小林	100
5	100005	张帆一	76
6	100005	张帆一	60
7	100004	赵明	56
8	100002	李红红	50
9	100001	张小林	35
10	100002	李红红	30

43、统计订购了商品的女客户的客户编号及其订单数量，不包含未订购商品的客户。

- 分析：查询结果信息来自 CustomerInfo、OrderList 和 GoodsInfo，故为多表查询；使用 JOIN 连接这三个表，再计算出每个订单金额。

- 设计的 SQL 语句：

```
SELECT A.客户编号,COUNT(*) AS '订单数量'
FROM CustomerInfo A JOIN OrderList B ON A.客户编号 = B.客户编号
WHERE A.性别 = '女'
GROUP BY A.客户编号
```

- 语句执行结果如下：



	客户编号	订单数量
1	100002	2
2	100006	1

44、统计各个女客户的客户编号及其订单数量，若客户未订购任何商品，则其订单数为 0，并按订单数量由高到低排序。

- 分析：上一题为自然连接，现在为左连接，再增加一个降序 DESC。

- 设计的 SQL 语句：

```
SELECT A.客户编号,COUNT(B.商品编号) AS '订单数量'
FROM CustomerInfo A LEFT JOIN OrderList B ON A.客户编号 = B.客户编号
WHERE A.性别 = '女'
GROUP BY A.客户编号
ORDER BY 订单数量 DESC
```

- 语句执行结果如下：



	客户编号	订单数量	
1	100002	2	
2	100006	1	
3	100003	0	

45、若客户下了 2 个及以上的食品类订单，请把客户编号及其食品类订单个数统计出来。

- 分析：查询结果信息来自 OrderList 和 GoodsInfo，故为多表查询；使用 JOIN 连接这两个表，再 WHERE 筛选。

- 设计的 SQL 语句：

```
SELECT A.客户编号,COUNT(*) AS '食品类订单个数'
FROM OrderList A JOIN GoodsInfo B ON A.商品编号 = B.商品编号
WHERE B.商品类别 = '食品'
GROUP BY A.客户编号
HAVING COUNT(*)>=2
```

- 语句执行结果如下：

结果 消息		
	客户编号	食品类订单数
1	100001	4

46、统计每个客户所下订单的总金额，未订购的客户不做统计，按总金额由大到小排序输出。

- 分析：查询结果信息来自 CustomerInfo、OrderList 和 GoodsInfo，故为多表查询；使用 JOIN 连接这三个表，再计算出每个订单金额并排序。
- 设计的 SQL 语句：

```
SELECT A.客户编号,SUM(B.数量* C.单价) AS '总金额'
FROM CustomerInfo A JOIN OrderList B ON A.客户编号 = B.客户编号
JOIN GoodsInfo C ON B.商品编号 = C.商品编号
GROUP BY A.客户编号
ORDER BY 总金额 DESC
```

- 语句执行结果如下：

结果 消息		
	客户编号	总金额
1	100001	435
2	100004	336
3	100006	175
4	100005	136
5	100002	80

47、统计客户编号及其所有订单的总金额，若客户未订购任何商品，则总金额为 0。按总金额由大到小排序输出。

- 分析：查询结果信息来自 CustomerInfo、OrderList 和 GoodsInfo，故为多表查询；使用 LEFT JOIN 连接这三个表，再计算出每个订单金额并排序。额外使用 ISNULL 函数。
- 设计的 SQL 语句：

```
SELECT A.客户编号,SUM(ISNULL(B.数量,0)*ISNULL(C.单价,0)) AS '总金额'
FROM CustomerInfo A
LEFT JOIN OrderList B ON A.客户编号 = B.客户编号
LEFT JOIN GoodsInfo C ON B.商品编号 = C.商品编号
GROUP BY A.客户编号
ORDER BY 总金额 DESC
```

- 语句执行结果如下：

	客户编号	总金额
1	100001	435
2	100004	336
3	100006	175
4	100005	136
5	100002	80
6	100003	0

48、统计客户所下订单的总金额，未订购的客户不做统计，按总金额由大到小排序输出前 3 个客户的编号及其总金额。

- 分析：在 46 题的基础上加上 Top(3)输出前三行。
- 设计的 SQL 语句：

```
SELECT Top(3) A.客户编号,SUM(B.数量*C.单价) AS '总金额'
FROM CustomerInfo A
JOIN OrderList B ON A.客户编号 = B.客户编号
JOIN GoodsInfo C ON B.商品编号 = C.商品编号
GROUP BY A.客户编号
ORDER BY 总金额 DESC
```

- 语句执行结果如下：



	客户编号	总金额
1	100001	435
2	100004	336
3	100006	175

四、实验总结

本次 SQL 实验以“GoodsOrder”数据库为基础，系统练习了多表关联查询、条件筛选、分组统计与空值处理等核心技能。通过对客户、订单与商品三张表的联合查询，我完成了 20 多个贴近实际的数据提取任务，提升了复杂查询的综合应用能力。

在题目 27 至 31 中，我掌握了多表连接与条件组合的运用方法，如 INNER JOIN 与 LEFT JOIN 的选择，以及如何通过条件筛选提取特定地区、类别或供应商的订单信息。题目 32 至 36 我联系了聚合函数与子查询的使用，包括订单金额计算、重复订购识别及连续下单判断等逻辑处理。37 至 40 题强化了我对外连接的理解，让我学会如何在统计时保留未下单客户的数据记录，并合理处理 NULL 值。在 41 至 48 题中，我进一步掌握了分组统计、排序输出及 TOP 筛选等操作。

总体而言，这次实验不仅夯实了我的 SQL 语法基础，也让我对一些查询语句更加了解熟练。