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 子查询和综合查询 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. 掌握嵌套查询的基本结构
2. 理解不相关子查询和相关子查询的区别
3. 掌握 IN 子查询、比较子查询、EXISTS 子查询的设计方法
4. 学习并练习公共表表达式(CTE)
5.本次实验主要练习 SELECT 综合查询的设计。

二、实验内容

(1) 教材附录中关于 StudentCourse 数据库的练习；

(2) 关于 GoodsOrder 数据库的查询设计题。**在 GoodsOrder 数据库中，用 SQL 语句实现如下查询：**

49、查询所在省市为“江苏南京”的客户的订单信息。

50、查询库存量超过平均库存量的商品信息。

51、查询所有订购了商品的客户的平均年龄。

52、查询使用“支付宝”方式订购的客户编号和姓名。

53、查询使用“支付宝”方式订购并且送货方式为“送货上门”的客户编号和姓名。

54、查询订购了“新新文化用品制造厂”产品的客户编号、客户姓名和相应订单信息。

55、查询订购了“新新文化用品制造厂”产品、且订购时间为“2020-02-19”及以后的客户编号、客户姓名和相应订单信息。

56、查询“食品”类商品被哪些省市的客户订购。

57、查询订购了“食品”类商品的客户编号和客户姓名。

58、查询订购了“食品”类商品的“江苏南京”的客户编号、姓名、年龄。

59、查询订购了“食品”类商品的“江苏南京”的客户的最小年龄。

60、查询订购了“食品”类商品的“江苏南京”的年龄最小的客户信息。

61、查询“张小林”未订购过的商品信息。

62、查询订购了“张小林”订购过的商品的客户编号和客户姓名。

63、查询订购数量比“王芳芳”所有订单数量都小的订单信息。

64、查询订购了库存量在 50 及以上的商品的客户编号和客户姓名。

65、查询年龄小于所有客户平均年龄的客户的订单信息。

66、查询具有相同出生日期的客户信息。

67、查询各生产商制造的商品被“江苏”客户订购的总数量、并由大到小排序，若生产商的商品未被“江苏”客户订购过，统计为 0。

68、假设用单个订单购物花费来评价客户购买力，找出“最能购买”（即单笔购物花费最多）的客户编号、客户姓名及其花费最多的订单金额。

69、查询有一个或一个以上订单金额大于 30 的所有客户的编号、姓名。

70、统计各个编号的商品被订购的总数量和总金额，若商品未被订购过，统计为 0。

71、统计各个类别的商品被订购的总数量和总金额，若商品未被订购过，统计为 0，按订购总数量由高到低输出。

72、“最具人气商品” ：查找被订购总数量最多的商品编号、商品名称及其订购总数

量。

73、“销量排行榜冠军”：找出被订购总数量最多的商品类别及其销量。

74、“分类销售额冠军排行榜” ：统计各类商品中销售总额最高的商品信息，输出其商品编号、类别、生产商及其销售总额，并由高到低排序 。

75、统计“30 岁以下”、“30~39(含 30)”、“40~49(含 40)”、“50 岁及以上”、“未知年龄”的客户数，若客户的出生日期为空，则统计在“未知年龄”段中。

76、查询至少有连续两天购买下单的客户编号和客户姓名。

77、现在需要统计客户有连续 2 天购物行为模式的比例，进行计算输出。

78、查询每个客户最长连续下单的天数，若未连续下单则输出“未连续下单”，若用户未购物，则输出“未购物”。

三、实验过程与结果

（1）StudentCourse 实验

1、查找与“丁一平”在同一年出生的学生情况。

- 分析：在 Student 表中嵌套查询，首先查询出姓名为“丁一平”的学生的出生年份，再查询与它相同出生年份的学生信息。
- 设计的 SQL 语句：

```
SELECT *
    FROM Student
    WHERE YEAR(出生时间) = (
        SELECT YEAR(出生时间)
        FROM Student
        WHERE  姓名  = '丁一平'
    )
```

- 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
1	070101	丁一平	计算机科学与技术	男	1989-05-01 00:00:00	80	三好生
2	070202	王波	电子信息工程	男	1989-02-18 00:00:00	76	多次获奖学金
3	070206	赵红涛	电子信息工程	男	1989-03-20 00:00:00	72	NULL
4	070208	李进	电子信息工程	男	1989-09-12 00:00:00	74	NULL

2、查询未选修任何课程的学生情况

- 分析：嵌套查询，先在 StuCourse 表里面查询出有选秀课程的学生的学生学号，再在 Student 表里面查询学号 NOT IN 里面的学生信息。
- 设计的 SQL 语句：

```
SELECT *
    FROM Student
    WHERE  学号  NOT IN (
        SELECT  学号
        FROM StuCourse
    )
```

- 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
1	070207	朱平平	电子信息工程	女	1990-01-10 00:00:00	74	NULL
2	070208	李进	电子信息工程	男	1989-09-12 00:00:00	74	NULL

3、查找选修了“电路基础”课程的学生学号和姓名。

- 分析：嵌套查询，用 EXISTS 谓词检测查询是否为空。
- 设计的 SQL 语句：

```
SELECT 学号, 姓名
    FROM Student x
    WHERE EXISTS
    (
        SELECT *
            FROM StuCourse a, Course b
            WHERE a.课程号=b.课程号 AND x.学号=a.学号 AND 课程名='电路基础'
    )
```

- 语句执行结果如下：

	学号	姓名
1	070201	王燕燕
2	070202	王波
3	070206	赵红涛

4、查找至少选修了学号为“070101”学生选修的全部课程的学生学号和姓名。

- 分析：通过嵌套查询的方式，查找那些选修了学号为 070101 学生所有课程的学生。首先提取 070101 选修的全部课程，然后用双重 NOT EXISTS 判断其他学生是否也选了这些课。只有当某学生没有遗漏任何一门课时，才会被保留在结果中，从而实现了“课程集合包含”的筛选逻辑。
- 设计的 SQL 语句：

```
SELECT 学号,姓名
    FROM Student
    WHERE 学号 IN
    (
        SELECT 学号
            FROM StuCourse x
            WHERE NOT EXISTS
            (
                SELECT *
                    FROM StuCourse y
                    WHERE y.学号='070101' AND NOT EXISTS
```

```
                    (
                        SELECT *
                            FROM StuCourse z
                            WHERE z.学号=x.学号  AND z.课程号  = y.课程号
                    )
                )
            )
```

- 语句执行结果如下：

	学号	姓名
1	070101	丁一平
2	070102	王红
3	070105	朱江
4	070206	赵红涛

5、查找未选修“程序设计基础”课程的学生情况。

- 分析：嵌套查询，先查询选修了这门课程的学生的学号，再查询 NOT IN 这些学号里面的学生。
- 设计的 SQL 语句：

```
SELECT *
    FROM Student
    WHERE  学号  NOT IN
    (
        SELECT  学号
            FROM StuCourse
            WHERE  课程号  =
            (
                SELECT  课程号
                    FROM Course
                    WHERE  课程名  = '程序设计基础'
            )
    )
```

- 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
1	070201	王燕燕	电子信息工程	女	1988-11-19 00:00:00	74	NULL
2	070202	王波	电子信息工程	男	1989-02-18 00:00:00	76	多次获奖学金
3	070207	朱平平	电子信息工程	女	1990-01-10 00:00:00	74	NULL
4	070208	李进	电子信息工程	男	1989-09-12 00:00:00	74	NULL

（2）GoodsOrder 实验

49、查询所在省市为“江苏南京”的客户的订单信息。

- 分析：嵌套查询，先在 CustomerInfo 里面查询所在省市为“江苏南京”的客户编号，再

在 OrderList 里面查询客户编号相等的订单信息。

- 设计的 SQL 语句如下：

```
SELECT *
    FROM OrderList A
    WHERE A.客户编号  IN
    (
        SELECT  客户编号
            FROM CustomerInfo
            WHERE  所在省市 ＝'江苏南京'
    )
```

- 语句执行结果如下：

结果 消息

	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	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	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）查询所在省市为“江苏南京”或“江苏苏州”的客户的订单信息如下：

```
SELECT *
    FROM OrderList A
    WHERE A.客户编号  IN
    (
        SELECT  客户编号
            FROM CustomerInfo
            WHERE  所在省市  IN ('江苏南京','江苏苏州')
    )
```

	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	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	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	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	10020001	2020-02-23 09:00:00.000	5	2020-02-26 00:00:00.000	信用卡	送货上门

（2）使用 JOIN 语句进行连接查询，如下：

```
SELECT A.*
    FROM OrderList A JOIN CustomerInfo B ON A.客户编号 ＝B.客户编号
    WHERE B.所在省市 ＝'江苏南京'
```

结果 消息

	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	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	10020001	2020-02-23 09:00:00.000	5	2020-02-26 00:00:00.000	信用卡	送货上门

50、查询库存量超过平均库存量的商品信息。

- 分析：先在 GoodsInfo 中用 AVG（）函数算出平均库存量，再查询库存量大于此的商

品信息。

- 设计的 SQL 语句：

```
SELECT *
    FROM GoodsInfo
    WHERE 库存量 >
    (
        SELECT AVG(库存量)
        FROM GoodsInfo
    )
```

- 语句执行结果如下：

结果 消息

	商品编号	商品类别	商品名称	品牌	单价	生产商	保质期	库存里	备注
1	10010001	食品	咖啡	宇一	50	宇一饮料公司	2021-08-31 00:00:00.000	100	NULL
2	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
3	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
4	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

- 进一步的思考与练习：

（1）查询单价超过平均单价的商品信息，如下：

```
SELECT *
    FROM GoodsInfo
    WHERE 单价 >
    (
        SELECT AVG(单价)
            FROM GoodsInfo
    )
```

结果 消息

	商品编号	商品类别	商品名称	品牌	单价	生产商	保质期	库存里	备注
1	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
2	20180002	服装	T恤	天天	120	天天服饰公司	2000-01-01 00:00:00.000	10	NULL
3	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
4	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

51、查询所有订购了商品的客户的平均年龄。

- 分析：先在 OrderList 和 CustomerInfo 表中查询出订购商品的客户编号，在计算出其平均年龄。
- 设计的 SQL 语句：

```
SELECT AVG(DATEDIFF(YEAR,出生日期,GetDate())) AS '平均年龄'
    FROM CustomerInfo A
    WHERE A.客户编号 IN
    (
        SELECT DISTINCT B.客户编号
            FROM OrderList B
            WHERE A.客户编号 = B.客户编号
    )
```

- 语句执行结果如下：

	平均年龄
1	34

- 进一步的思考与练习：

（1）查询所有订购了商品的客户的最大年龄和最小年龄，如下：

```
SELECT MAX(DATEDIFF(YEAR, 出生日期 ,GetDate())) AS '最大年龄',MIN(DATEDIFF(YEAR,出生日期,GetDate())) AS '最小年龄'
    FROM CustomerInfo A
    WHERE A.客户编号 IN
    (
        SELECT DISTINCT B.客户编号
            FROM OrderList B
            WHERE A.客户编号 =B.客户编号
    )
```

结果 消息

	最大年龄	最小年龄
1	43	29

52、查询使用“支付宝”方式订购的客户编号和姓名。

- 分析：嵌套查询，现在 OrderList 表中查询支付方式为“支付宝”的客户编号，再在 CustomerInfo 表中查询在这个客户编号集合中的客户编号和姓名。
- 设计的 SQL 语句：

```
SELECT 客户编号,客户姓名
    FROM CustomerInfo A
    WHERE A.客户编号 IN
    (
        SELECT DISTINCT B.客户编号
            FROM OrderList B
            WHERE B.付款方式 ='支付宝'
    )
```

- 语句执行结果如下：

结果 消息

	客户编号	客户姓名
1	100001	张小林
2	100005	张帆一

- 进一步的思考与练习：

（1）查询送货方式为送货上门的客户编号和姓名，如下：

```
SELECT  客户编号,客户姓名
    FROM CustomerInfo A
    WHERE A.客户编号  IN
    (
        SELECT DISTINCT B.客户编号
            FROM OrderList B
            WHERE B.送货方式 ='送货上门'
    )
```

结果 消息

	客户编号	客户姓名
1	100001	张小林
2	100004	赵明
3	100005	张帆一
4	100006	王芳芳

53、查询使用“支付宝”方式订购并且送货方式为“送货上门”的客户编号和姓名。

- 分析：在上一个查询的基础上的子层查询中添加条件即可。
- 设计的 SQL 语句：

```
SELECT  客户编号,客户姓名
    FROM CustomerInfo A
    WHERE A.客户编号  IN
    (
        SELECT DISTINCT B.客户编号
            FROM OrderList B
            WHERE B.送货方式 ='送货上门' AND B.付款方式 ='支付宝'
    )
```

- 语句执行结果如下：

结果 消息

	客户编号	客户姓名
1	100005	张帆一

- 进一步的思考与练习：

（1）另一种方法，如下：

```
SELECT DISTINCT A.客户编号,A.客户姓名
    FROM CustomerInfo A JOIN OrderList B ON A.客户编号 =B.客户编号
    WHERE B.送货方式 ='送货上门' AND B.付款方式 ='支付宝'
```

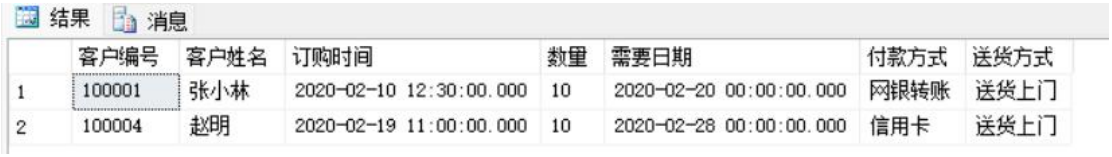
54、查询订购了“新新文化用品制造厂”产品的客户编号、客户姓名和相应订单信息。

- 分析：先在 GoodsInfo 表中筛选出生产商为“新新文化用品制造厂”的商品编号的集合，再将 OrderList 表和 CustomerInfo 表连接起来，进行查询。

- 设计的 SQL 语句：

```
SELECT A.客户编号, A.客户姓名, 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
    JOIN OrderList B ON A.客户编号 = B.客户编号
    WHERE B.商品编号 IN
    (
        SELECT 商品编号
            FROM GoodsInfo C
            WHERE C.生产商 = '新新文化用品制造厂'
    )
```

- 语句执行结果如下：



结果 消息

	客户编号	客户姓名	订购时间	数量	需要日期	付款方式	送货方式
1	100001	张小林	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	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）另一种方法：

```
SELECT A.客户编号, A.客户姓名, 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
        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
    WHERE C.生产商 = '新新文化用品制造厂'
```

55、查询订购了“新新文化用品制造厂”产品、且订购时间为“2020-02-19”及以后的客户编号、客户姓名和相应订单信息。

- 分析：在上一个查询的基础上加上订购时间的选择。
- 设计的 SQL 语句：

```
SELECT A.客户编号, A.客户姓名, 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
    JOIN OrderList B ON A.客户编号 = B.客户编号
    WHERE B.商品编号 IN
    (
        SELECT 商品编号
            FROM GoodsInfo C
            WHERE C.生产商 = '新新文化用品制造厂'
    )
    AND 订购时间 >= '2020-02-19'
```

- 语句执行结果如下：

	客户编号	客户姓名	订购时间	数量	需要日期	付款方式	送货方式
1	100004	赵明	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 EXISTS 语句方法，如下：

```
SELECT A.客户编号, A.客户姓名, 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
        JOIN OrderList B ON A.客户编号 = B.客户编号
        WHERE EXISTS (
            SELECT 1
            FROM GoodsInfo C
            WHERE C.商品编号 = B.商品编号
            AND C.生产商 = '新新文化用品制造厂'
        )
    AND B.订购时间 >= '2020-02-19';
```

56、查询“食品”类商品被哪些省市的客户订购。

- 分析：嵌套查询，先在 GoodsInfo 商品中筛选出商品类别为“食品”类别的商品编号，再在 OrderList 表中根据此商品编号筛选出客户编号，再在 CustomerInfo 表中筛选出对应省市。
- 设计的 SQL 语句：

```
SELECT A.所在省市
    FROM CustomerInfo A
    WHERE A.客户编号 IN
    (
        SELECT B.客户编号
            FROM OrderList B
            WHERE B.商品编号 IN
            (
                SELECT C.商品编号
                    FROM GoodsInfo C
                    WHERE 商品类别 = '食品'
            )
    )
```

- 语句执行结果如下：

	所在省市
1	江苏南京
2	江苏苏州
3	江苏南京

- 进一步的思考与练习：

（1）使用 EXISTS 写法，如下：

```
SELECT A.所在省市
    FROM CustomerInfo A
    WHERE EXISTS
```

```
(
    SELECT 1
    FROM OrderList B
    JOIN GoodsInfo C ON B.商品编号 ＝C.商品编号
    WHERE B.客户编号 ＝A.客户编号 AND C.商品类别 ＝'食品'
)
```

57、查询订购了“食品”类商品的客户编号和客户姓名。

- 分析：与上一个查询类似，在最开始的查询中选择客户编号和客户姓名两列。
- 设计的 SQL 语句：

```
SELECT A.客户编号,A.客户姓名
        FROM CustomerInfo A
        WHERE A.客户编号 IN
        (
            SELECT B.客户编号
                FROM OrderList B
                WHERE B.商品编号 IN
                (
                    SELECT C.商品编号
                        FROM GoodsInfo C
                        WHERE 商品类别 ＝'食品'
                )
        )
```

- 语句执行结果如下：

结果 消息

	客户编号	客户姓名
1	100001	张小林
2	100002	李红红
3	100006	王芳芳

- 进一步的思考与练习：

（1）使用 EXISTS 方法，如下：

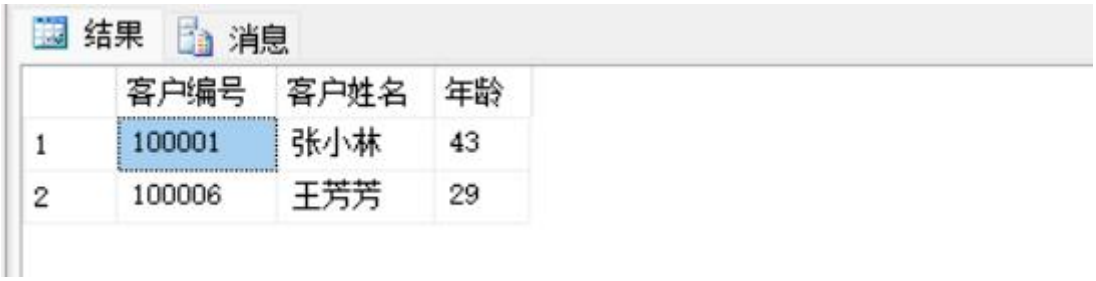
```
SELECT A.客户编号,A.客户姓名
    FROM CustomerInfo A
    WHERE EXISTS
    (
        SELECT 1
        FROM OrderList B
        JOIN GoodsInfo C ON B.商品编号 ＝C.商品编号
        WHERE B.客户编号 ＝A.客户编号 AND C.商品类别 ＝'食品'
    )
```

58、查询订购了“食品”类商品的“江苏南京”的客户编号、姓名、年龄。

- 分析：在上上个查询的基础上增加条件，选择对应此查询的目标列。年龄用 DateDIFF 函数计算。
- 设计的 SQL 语句：

```
SELECT A.客户编号,A.客户姓名,DATEDIFF(YEAR,A.出生日期,GETDATE()) AS '年龄'
        FROM CustomerInfo A
        WHERE A.客户编号  IN
        (
            SELECT B.客户编号
                FROM OrderList B
                WHERE B.商品编号  IN
                (
                    SELECT C.商品编号
                        FROM GoodsInfo C
                        WHERE  商品类别 ＝'食品'
                )
        ) AND  所在省市 ＝'江苏南京'
```

- 语句执行结果如下：



	客户编号	客户姓名	年龄
1	100001	张小林	43
2	100006	王芳芳	29

- 进一步的思考与练习：

（1）使用 EXISTS 方法，如下：

```
SELECT A.客户编号,A.客户姓名,DATEDIFF(YEAR,A.出生日期,GETDATE()) AS '年龄'
        FROM CustomerInfo A
        WHERE EXISTS
        (
            SELECT 1
            FROM OrderList B
            JOIN GoodsInfo C ON B.商品编号 ＝C.商品编号
            WHERE B.客户编号 ＝A.客户编号  AND C.商品类别 ＝'食品'
        ) AND  所在省市 ＝'江苏南京'
```

59、查询订购了“食品”类商品的“江苏南京”的客户的最小年龄。

- 分析：与上一个查询类似，用 MIN（）函数。
- 设计的 SQL 语句：

```
SELECT MIN(DATEDIFF(YEAR,A.出生日期,GETDATE())) AS '最小年龄'
        FROM CustomerInfo A
        WHERE A.客户编号  IN
```

```
(
    SELECT B.客户编号
        FROM OrderList B
        WHERE B.商品编号  IN
        (
            SELECT C.商品编号
                FROM GoodsInfo C
                WHERE  商品类别 ='食品'
        )
) AND  所在省市 ='江苏南京'
```

- 语句执行结果如下：

	最小年龄
1	29

- 进一步的思考与练习：

（1）使用 EXISTS 方法，如下：

```
SELECT MIN(DATEDIFF(YEAR,A.出生日期,GETDATE())) AS '最小年龄'
        FROM CustomerInfo A
        WHERE EXISTS
        (
            SELECT 1
            FROM OrderList B
            JOIN GoodsInfo C ON B.商品编号 = C.商品编号
            WHERE B.客户编号 = A.客户编号  AND C.商品类别 ='食品'
        ) AND  所在省市 ='江苏南京'
```

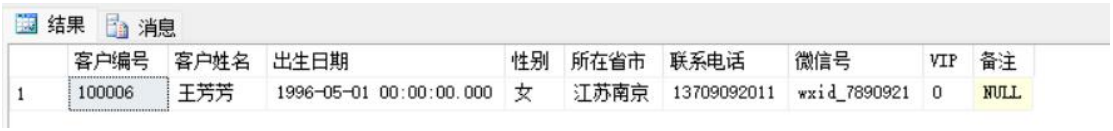
60、查询订购了“食品”类商品的“江苏南京”的年龄最小的客户信息。

- 分析：在上一个查询中嵌套一个查询，目标为 CustomerInfo 所有列。
- 设计的 SQL 语句：

```
SELECT D.*
FROM CustomerInfo D
WHERE DATEDIFF(YEAR,D.出生日期,GETDATE()) =
(
    SELECT MIN(DATEDIFF(YEAR,A.出生日期,GETDATE())) AS '最小年龄'
        FROM CustomerInfo A
        WHERE A.客户编号  IN
        (
            SELECT B.客户编号
                FROM OrderList B
                WHERE B.商品编号  IN
                (
                    SELECT C.商品编号
                        FROM GoodsInfo C
```

```
                    WHERE 商品类别 ='食品'
                )
            ) AND 所在省市 ='江苏南京'
    )
```

- 语句执行结果如下：



结果 消息

	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	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 EXISTS 方法，如下：

```
SELECT D.*
FROM CustomerInfo D
WHERE DATEDIFF(YEAR,D.出生日期,GETDATE()) =
(
SELECT MIN(DATEDIFF(YEAR,A.出生日期,GETDATE())) AS '最小年龄'
        FROM CustomerInfo A
        WHERE EXISTS
        (
            SELECT 1
            FROM OrderList B
            JOIN GoodsInfo C ON B.商品编号 =C.商品编号
            WHERE B.客户编号 =A.客户编号 AND C.商品类别 ='食品'
        ) AND 所在省市 ='江苏南京'
)
```

61、查询“张小林”未订购过的商品信息。

- 分析：嵌套查询，先按客户编号连接 CustomerInfo 表和 OrderList 表，再在其中条件判断用户姓名和商品编号，再在 GoodsInfo 表查询不在此编号中的商品信息，目标为此表的全部列。
- 设计的 SQL 语句：

```
SELECT D.*
    FROM GoodsInfo D
    WHERE NOT EXISTS
    (
        SELECT 商品编号
            FROM OrderList A
            JOIN CustomerInfo B ON A.客户编号 =B.客户编号
            WHERE A.商品编号 =D.商品编号 AND B.客户姓名 ='张小林'
    )
```

- 语句执行结果如下：

结果 消息

	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
1	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
2	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
3	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
4	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
5	20180002	服装	T恤	天天	120	天天服饰公司	2000-01-01 00:00:00.000	10	NULL
6	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
7	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
8	40010002	图书	豆浆的做法	新华	20	新华图书出版公司	2000-01-01 00:00:00.000	20	NULL
9	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
10	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
11	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

62、查询订购了“张小林”订购过的商品的客户编号和客户姓名。

- 分析：在上一个查询的基础上，将 NOT EXISTS 换成 EXISTS，再在 CustomerInfo 表和 OrderList 表中查询对应的客户编号和客户姓名。
- 设计的 SQL 语句：

```
SELECT DISTINCT A.客户编号, B.客户姓名
    FROM OrderList A JOIN CustomerInfo B ON A.客户编号 = B.客户编号
    WHERE EXISTS(
        SELECT 1
            FROM OrderList C JOIN CustomerInfo D ON C.客户编号 = D.客户编号
            WHERE D.客户姓名 = '张小林' AND C.商品编号 = A.商品编号
    ) AND B.客户姓名 != '张小林'
```

- 语句执行结果如下：

消息

	客户编号	客户姓名
1	100002	李红红

63、查询订购数量比“王芳芳”所有订单数量都小的订单信息。

- 分析：先要查询客户姓名为“王芳芳”的所有订单，再算出其所有订单数量，再在 OrderList 里查询比较得到订单信息。
- 设计的 SQL 语句：

```
SELECT A.*
    FROM OrderList A
    WHERE A.数量<
    (
        SELECT SUM(数量) AS '订单数量'
            FROM OrderList B
            WHERE B.客户编号 =
            (
                SELECT C.客户编号
```

```
                FROM CustomerInfo C
                WHERE C.客户姓名 ='王芳芳'
            )
        )
```

- 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	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	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
4	100004	20180002	2020-02-19 10:00:00.000	1	2020-02-28 00:00:00.000	信用卡	送货上门
5	100004	50020002	2020-02-19 10:40:00.000	2	2020-02-28 00:00:00.000	信用卡	送货上门
6	100005	40010001	2020-02-20 08:00:00.000	2	2020-02-27 00:00:00.000	支付宝	送货上门
7	100005	40010002	2020-02-20 08:20:00.000	3	2020-02-27 00:00:00.000	支付宝	送货上门

- 进一步的思考与练习：

（1）查询订购数量比“张帆一”所有订单数量都多的订单信息，如下：

```
SELECT A.*
    FROM OrderList A
    WHERE A.数量>
    (
        SELECT SUM(数量) AS '订单数量'
            FROM OrderList B
            WHERE B.客户编号 =
            (
                SELECT C.客户编号
                    FROM CustomerInfo C
                    WHERE C.客户姓名 ='张帆一'
            )
    )
```

结果 消息

	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	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	30010002	2020-02-19 11:00:00.000	10	2020-02-28 00:00:00.000	信用卡	送货上门

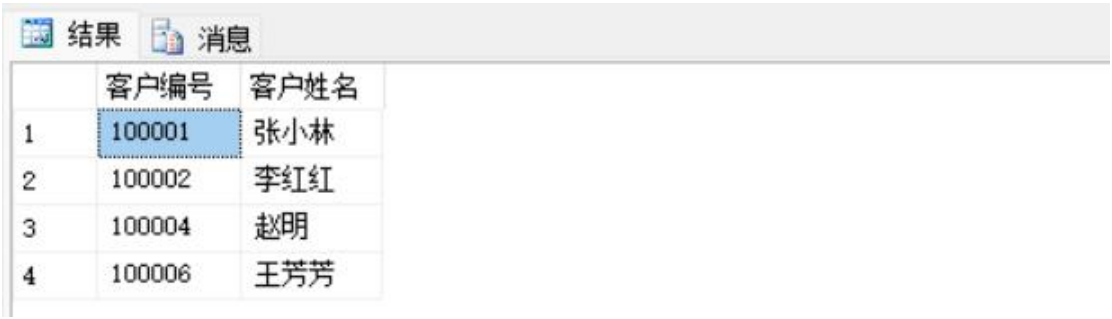
64、查询订购了库存量在 50 及以上的商品的客户编号和客户姓名。

- 分析：嵌套查询，先根据 GoodsInfo 表中的库存量查询库存量大于等于 50 的商品编号，再在 OrderList 表中根据对应商品编号查询满足条件的客户编号，再联合 CustomerInfo 表查询客户编号和客户姓名。
- 设计的 SQL 语句：

```
SELECT A.客户编号,A.客户姓名
    FROM CustomerInfo A
    WHERE A.客户编号  IN
    (
        SELECT B.客户编号
            FROM OrderList B
```

```
        WHERE B.商品编号  IN
        (
            SELECT C.商品编号
                FROM GoodsInfo C
                WHERE C.库存量  >= 50
        )
    )
```

- 语句执行结果如下：



结果 消息

	客户编号	客户姓名
1	100001	张小林
2	100002	李红红
3	100004	赵明
4	100006	王芳芳

- 进一步的思考与练习：

（1）使用 EXISTS 方法，如下：

```
SELECT A.客户编号, A.客户姓名
    FROM CustomerInfo A
    WHERE EXISTS (
        SELECT 1
            FROM OrderList B
            JOIN GoodsInfo C ON B.商品编号  = C.商品编号
            WHERE B.客户编号  = A.客户编号
        AND C.库存量  >= 50
    )
```

65、查询年龄小于所有客户平均年龄的客户的订单信息。

- 分析：先利用 CustomerInfo 表算出所有客户平均年龄，再在此表中得到客户编号，再在 OrderList 表中进行查询，实现嵌套查询。
- 设计的 SQL 语句：

```
SELECT A.*
    FROM OrderList A
    WHERE A.客户编号  IN
    (
        SELECT C.客户编号
            FROM CustomerInfo C
            WHERE DATEDIFF(YEAR,C.出生日期,GETDATE()) <
            (
                SELECT AVG(DATEDIFF(YEAR,B.出生日期,GETDATE())) AS '
平均年龄'
                    FROM CustomerInfo B
            )
```

)

- 语句执行结果如下：

结果 消息

	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	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
2	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
3	100004	20180002	2020-02-19 10:00:00.000	1	2020-02-28 00:00:00.000	信用卡	送货上门
4	100004	30010002	2020-02-19 11:00:00.000	10	2020-02-28 00:00:00.000	信用卡	送货上门
5	100004	50020002	2020-02-19 10:40:00.000	2	2020-02-28 00:00:00.000	信用卡	送货上门
6	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 EXISTS 方法，如下：

```
SELECT A.*
    FROM OrderList A
    WHERE A.客户编号  IN
    (
        SELECT C.客户编号
            FROM CustomerInfo C
            WHERE DATEDIFF(YEAR,C.出生日期,GETDATE()) <
            (
                SELECT AVG(DATEDIFF(YEAR,B.出生日期,GETDATE())) AS '
平均年龄'
                FROM CustomerInfo B
        )
    )
```

66、查询具有相同出生日期的客户信息。

- 分析：先使用 GROUP BY 和 HAVING 子句在 CustomerInfo 表中查找出出生日期出现超过一次的用户，再在 CustomerInfo 语句中再次选择判断。
- 设计的 SQL 语句：

```
SELECT A.*
    FROM CustomerInfo A
    WHERE A.出生日期  IN
    (
        SELECT B.出生日期
            FROM CustomerInfo B
            GROUP BY B.出生日期
            HAVING COUNT(*)>1
    )
```

- 语句执行结果如下：

结果 消息

	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	100003	王晓美	1986-08-20 00:00:00.000	女	上海市	02166552101	wxid_0021001	0	新客户
2	100004	赵明	1992-03-28 00:00:00.000	男	河南郑州	13809900118	NULL	0	新客户
3	100009	张萍萍	1986-08-20 00:00:00.000	女	山东青岛	18000990811	wxid_9901	0	NULL
4	100010	赵强	1992-03-28 00:00:00.000	男	河南郑州	13268120812	NULL	0	NULL

- 进一步的思考与练习：

（1）另一种方法，如下：

```
SELECT A.*
    FROM CustomerInfo A JOIN CustomerInfo B
    ON A.客户编号!=B.客户编号  AND A.出生日期=B.出生日期
```

67、查询各生产商制造的商品被“江苏”客户订购的总数量、并由大到小排序，若生产商的商品未被“江苏”客户订购过，统计为 0。

- 分析：先将 GoodsInfo 表与 OrderList 表和 CustomerInfo 表根据题目条件左连接，再根据生产商进行分组，并统计总数量。
- 设计的 SQL 语句：

```
    SELECT A.生产商,ISNULL(SUM(CASE WHEN C.所在省市  LIKE '江苏%' THEN D.
    数量  ELSE 0 END), 0) AS  总数量
        FROM GoodsInfo A
        LEFT JOIN OrderList D ON A.商品编号  = D.商品编号
        LEFT JOIN CustomerInfo C ON D.客户编号  = C.客户编号
        GROUP BY A.生产商
        ORDER BY  总数量  DESC;
```

- 语句执行结果如下：

结果 消息

	生产商	总数量
1	新新文化用品制造厂	10
2	宇一饮料公司	9
3	健康粮食生产基地	5
4	美好体育用品公司	1
5	天天服饰公司	0
6	新华图书出版公司	0

- 进一步的思考与练习：

（1）使用嵌套查询方法，如下：

```
SELECT T.生产商, SUM(T.江苏订单数量) AS '总数量'
    FROM
    (
        SELECT G.生产商,ISNULL((
            SELECT SUM(O.数量)
                FROM OrderList O
```

```
                JOIN CustomerInfo C ON O.客户编号 = C.客户编号
                WHERE O.商品编号 = G.商品编号 AND C.所在省市 LIKE '江
苏%'
            ),0) AS 江苏订单数量
            FROM GoodsInfo G
        ) T
        GROUP BY T.生产商
        ORDER BY 总数量 DESC
```

68、假设用单个订单购物花费来评价客户购买力，找出“最能购买”（即单笔购物花费最多）的客户编号、客户姓名及其花费最多的订单金额。

- 分析：先找出每个每个订单的金额，再降序排序，然后使用 TOP（1）选择输出第一行，即得到结果。
- 设计的 SQL 语句：

```
SELECT TOP(1) A.客户编号,C.客户姓名,(A.数量*B.单价) AS '订单金额'
    FROM OrderList A JOIN GoodsInfo B ON A.商品编号 = B.商品编号
    JOIN CustomerInfo C ON A.客户编号 = C.客户编号
    ORDER BY 订单金额 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

- 进一步的思考与练习：

（1）FROM 派生表+连接查询：

```
SELECT X.客户编号,Cust.客户姓名,X.订单金额
    FROM CustomerInfo Cust,
    (
        SELECT a.客户编号,数量*单价 AS '订单金额'
            FROM OrderList a,GoodsInfo b
            WHERE a.商品编号 = b.商品编号
    )X,
    (
        SELECT MAX(数量*单价) AS MaxCost
            FROM OrderList a,GoodsInfo b
            WHERE a.商品编号 = b.商品编号
    )Y
    WHERE X.客户编号 = Cust.客户编号 AND X.订单金额 = Y.MaxCost
```

（2）FROM 派生表+IN 子查询+连接查询：

```
SELECT Cust.客户编号,Cust.客户姓名,Z.订单金额
    FROM CustomerInfo Cust,
    (
```

```
        SELECT X.客户编号,X.订单金额
            FROM
            (
                SELECT  客户编号,数量*单价  AS  订单金额
                    FROM OrderList a,GoodsInfo b
                    WHERE a.商品编号 = b.商品编号
            )X
            WHERE X.订单金额  IN
            (
                SELECT MAX(数量*单价) AS MaxCost
                    FROM OrderList a,GoodsInfo b
                    WHERE a.商品编号 = b.商品编号
            )
    )Z
    WHERE Cust.客户编号 = Z.客户编号
```

（3）CTE 改造方案：

```
WITH X AS
    (
        SELECT a.客户编号,数量*单价  AS  订单金额
            FROM OrderList a,GoodsInfo b
            WHERE a.商品编号 = b.商品编号
    )
    SELECT X.客户编号,Cust.客户姓名,X.订单金额
        FROM CustomerInfo Cust,X,
            (
            SELECT MAX(订单金额) AS MaxCost
                FROM X
            ) Y
        WHERE X.客户编号 = Cust.客户编号  AND X.订单金额 = Y.MaxCost
```

69、查询有一个或一个以上订单金额大于 30 的所有客户的编号、姓名。

- 分析：先查询满足条件的客户编号，再在 CustomerInfo 表中查询对应姓名，通过去重即可得到查询目标。
- 设计的 SQL 语句：

```
WITH X AS
(
    SELECT c.客户编号,(单价*数量) AS  订单金额
        FROM GoodsInfo b,OrderList c
        WHERE b.商品编号 = c.商品编号
)
SELECT DISTINCT a.客户编号,a.客户姓名
    FROM X JOIN CustomerInfo a ON a.客户编号 = X.客户编号
    WHERE X.订单金额 > 30
```

- 语句执行结果如下：

	客户编号	客户姓名
1	100001	张小林
2	100002	李红红
3	100004	赵明
4	100005	张帆一
5	100006	王芳芳

- 进一步的思考与练习：

（1）使用 IN 方法，如下：

```
SELECT a.客户编号,a.客户姓名
    FROM CustomerInfo a
    WHERE a.客户编号 IN
    (
        SELECT c.客户编号
            FROM GoodsInfo b,OrderList c
            WHERE b.商品编号 = c.商品编号 AND 单价*数量 > 30
    )
```

（2）使用 EXISTS 方法，如下：

```
SELECT a.客户编号,a.客户姓名
    FROM CustomerInfo a
    WHERE EXISTS
    (
        SELECT 1
            FROM GoodsInfo b JOIN OrderList c ON b.商品编号 = c.商品编号
            WHERE c.客户编号 = a.客户编号 AND b.单价*c.数量 > 30
    )
```

70、统计各个编号的商品被订购的总数量和总金额，若商品未被订购过，统计为 0。

- 分析：将 GoodsInfo 表和 OrderList 表左连接，由 GoodsInfo 表的商品编号进行分组，使用 SUM（）函数计算总数量和总金额，在统计总数量时注意使用 ISNULL（）函数将 NULL 转换为 0。
- 设计的 SQL 语句：

```
SELECT a.商品编号,SUM(ISNULL(b.数量,0)) AS 订购总数量, SUM (a.单价 * ISNULL(b.数量,0)) AS 订购总金额
    FROM GoodsInfo a LEFT JOIN OrderList b ON a.商品编号 = b.商品编号
    GROUP BY a.商品编号
```

- 语句执行结果如下：

结果 消息

	商品编号	订购总数量	订购总金额
1	10010001	9	450
2	10010002	0	0
3	10020001	5	175
4	10020002	0	0
5	20180001	0	0
6	20180002	1	120
7	30010001	10	35
8	30010002	10	56
9	40010001	2	76
10	40010002	3	60
11	50020001	1	30
12	50020002	2	160
13	50020003	0	0

71、统计各个类别的商品被订购的总数量和总金额，若商品未被订购过，统计为 0，按订购总数量由高到低输出。

- 分析：与上一个查询类似，按类别进行分组，并增加排序函数。
- 设计的 SQL 语句：

```
SELECT a.商品类别,SUM(ISNULL(b.数量,0)) AS 订购总数量, SUM (a.单价 * ISNULL(b.数量,0)) AS 订购总金额
    FROM GoodsInfo a LEFT JOIN OrderList b ON a.商品编号 = b.商品编号
    GROUP BY a.商品类别
    ORDER BY 订购总数量 DESC
```

- 语句执行结果如下：

结果 消息

	商品类别	订购总数量	订购总金额
1	文具	20	91
2	食品	14	625
3	图书	5	136
4	体育用品	3	190
5	服装	1	120
6	家具	0	0

72、“最具人气商品”：查找被订购总数量最多的商品编号、商品名称及其订购总数量。

- 分析：先用公共表达式定义子表 X，目标列为商品编号及其订购总数量；再定义一个子表 Y 为最大订购数量，在进行查询。
- 设计的 SQL 语句：

```
WITH X AS
(
    SELECT a.商品编号,SUM(b.数量) AS  订购总数量
        FROM GoodsInfo a JOIN OrderList b ON a.商品编号 = b.商品编号
        GROUP BY a.商品编号
)
SELECT X.商品编号,c.商品名称,X.订购总数量
    FROM GoodsInfo c,X,
    (
        SELECT MAX(订购总数量) AS MaxCount
            FROM X
    )Y
    WHERE X.订购总数量 = Y.MaxCount AND X.商品编号 = c.商品编号
```

- 语句执行结果如下：

结果 消息

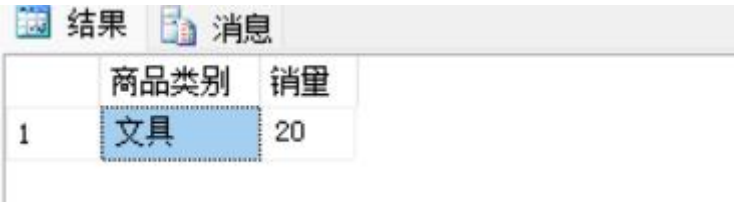
	商品编号	商品名称	订购总数量
1	30010001	签字笔	10
2	30010002	文件夹	10

73、“销量排行榜冠军”：找出被订购总数量最多的商品类别及其销量。

- 分析：与上一个查询类似，定义 X 为商品类别及销量也就是上一个查询中的订购总数量；后续操作与上一个查询相同。
- 设计的 SQL 语句：

```
WITH X AS
(
    SELECT a.商品类别,SUM(b.数量) AS  销量
        FROM GoodsInfo a JOIN OrderList b ON a.商品编号 = b.商品编号
        GROUP BY a.商品类别
)
SELECT X.商品类别,X.销量
    FROM X,
    (
        SELECT MAX(X.销量) AS MaxCount
            FROM X
    )Y
    WHERE X.销量 = Y.MaxCount
```

- 语句执行结果如下：



	商品类别	销量
1	文具	20

74、“分类销售额冠军排行榜”：统计各类商品中销售总额最高的商品信息，输出其商品编号、类别、生产商及其销售总额，并由高到低排序。

- 分析：先用公共表达式定义子表 X，目标列为商品类别、商品编号、生产商及其销售额；再定义一个子查询 Y，目标列为商品类别，该类最高销售额，再联合 X 表和 CustomerInfo 表进行查询。
- 设计的 SQL 语句：

```
WITH X AS
(
    SELECT a.商品类别,a.商品编号,SUM(b.数量*a.单价) AS  销量额
        FROM GoodsInfo a JOIN OrderList b ON a.商品编号  = b.商品编号
        GROUP BY a.商品类别,a.商品编号
)
SELECT c.商品编号,c.商品类别,c.生产商,Y.该类最高销售额
    FROM GoodsInfo c,X,
    (
        SELECT X.商品类别,MAX(X.销量额) AS  该类最高销售额
            FROM X
            GROUP BY X.商品类别
    ) Y
    WHERE X.销量额  =Y.该类最高销售额  AND X.商品编号  = c.商品编号
    ORDER BY  该类最高销售额  DESC
```

- 语句执行结果如下：

	商品编号	商品类别	生产商	该类最高销售额
1	10010001	食品	宇一饮料公司	450
2	50020002	体育用品	美好体育用品公司	160
3	20180002	服装	天天服饰公司	120
4	40010001	图书	新华图书出版公司	76
5	30010002	文具	新新文化用品制造厂	56

75、统计“30 岁以下”、“30~39(含 30)”、“40~49(含 40)”、“50 岁及以上”、“未知年龄”的客户数，若客户的出生日期为空，则统计在“未知年龄”段中。

- 分析：使用 CASE 语句进行条件判断。
- 设计的 SQL 语句：

```
SELECT  年龄段,COUNT(*) AS  客户数
```

```
FROM
(
    SELECT
        CASE
            WHEN c.出生日期  IS NULL THEN '未知年龄'
            WHEN DATEDIFF(YEAR,c.出生日期,GETDATE())<30 THEN '30 岁以下'
            WHEN DATEDIFF(YEAR,c.出生日期,GETDATE()) BETWEEN 30 AND 39 THEN '30~39 岁'
            WHEN DATEDIFF(YEAR,c.出生日期,GETDATE()) BETWEEN 40 AND 49 THEN '40~49 岁'
            WHEN DATEDIFF(YEAR,c.出生日期,GETDATE())>=50 THEN '50 岁及以上'
        END AS  年龄段
        FROM CustomerInfo c
) AS age
GROUP BY  年龄段
ORDER BY  客户数  DESC
```

- 语句执行结果如下：

结果 消息

	年龄段	客户数
1	30~39岁	7
2	30岁以下	4
3	50岁及以上	2
4	未知年龄	2
5	40~49岁	1

76、查询至少有连续两天购买下单的客户编号和客户姓名。

- 分析：先用公共表达式定义子表 X，目标列为客户编号，客户姓名，订购日期和按客户编号进行分组，再按订购日期进行升序排序的排序号；接着 X 表自连接，连接条件为 a 表的序号加 1 等于 b 表的序号，在通过 DATEDIFF（）函数判断，得到查询结果。
- 设计的 SQL 语句：

```
WITH X AS
(
    SELECT a.客户编号,b.客户姓名,CAST(a.订购时间  AS DATE) AS  订购日期,
        ROW_NUMBER() OVER
        (
            PARTITION BY a.客户编号
            ORDER BY CAST(a.订购时间  AS DATE)
         ) AS  序号
        FROM OrderList a JOIN CustomerInfo b ON a.客户编号  = b.客户编号
)
SELECT DISTINCT a.客户编号,a.客户姓名
```

```
    FROM X a JOIN X b ON a.客户编号 = b.客户编号 AND b.序号 = a.序号+1
    WHERE DATEDIFF(DAY,a.订购日期,b.订购日期) = 1
```

- 语句执行结果如下：

	客户编号	客户姓名
1	100001	张小林
2	100002	李红红

77、现在需要统计客户有连续 2 天购物行为模式的比例，进行计算输出。

- 分析：在上一个查询的基础上添加满足条件的订单数子表 Y 和总订单数子表 Z，最后选择两个相除，注意把 Y 的数利用 CAST（）函数换成 FLOAT 类型，不然结果为 0。
- 设计的 SQL 语句：

```
WITH X AS
(
    SELECT a.客户编号,b.客户姓名,CAST(a.订购时间 AS DATE) AS 订购日期,
        ROW_NUMBER() OVER
        (
            PARTITION BY a.客户编号
            ORDER BY CAST(a.订购时间 AS DATE)
        ) AS 序号
        FROM OrderList a JOIN CustomerInfo b ON a.客户编号 = b.客户编号
),
Y AS(
    SELECT COUNT(*) AS 目标次数
        FROM X a JOIN X b ON a.客户编号 = b.客户编号 AND b.序号 = a.序号+1
        WHERE DATEDIFF(DAY,a.订购日期,b.订购日期) = 1
),
Z AS(
    SELECT COUNT(*) AS 总次数
        FROM OrderList
)
SELECT ROUND(CAST(Y.目标次数 AS FLOAT) /Z.总次数,2) AS schema_rate
    FROM Y,Z
```

- 语句执行结果如下：

	schema_rate
1	0.31

78 、查询每个客户最长连续下单的天数，若未连续下单则输出“未连续下单”，若用

户未购物，则输出“未购物”。

- 分析：分别求出 X，连续分组，连续天数统计，最长连续天数，所有客户子表；再加 CASE 函数判断。
- 设计的 SQL 语句：

```
WITH X AS
(
    SELECT a.客户编号,b.客户姓名,CAST(a.订购时间 AS DATE) AS 订购日期,
        ROW_NUMBER() OVER
        (
            PARTITION BY a.客户编号
            ORDER BY CAST(a.订购时间 AS DATE)
        ) AS 序号
        FROM OrderList a JOIN CustomerInfo b ON a.客户编号 = b.客户编号
),
连续分组 AS(
    SELECT 客户编号,客户姓名,订购日期,序号,DATEADD(DAY, -序号, 订购日期) AS 分组键
        FROM X
),
连续天数统计 AS(
    SELECT 客户编号,客户姓名,COUNT(*) AS 连续天数
        FROM 连续分组
        GROUP BY 客户编号,客户姓名,分组键
),
最长连续天数 AS(
    SELECT 客户编号,客户姓名,MAX(连续天数) AS 最长连续天数
        FROM 连续天数统计
        GROUP BY 客户编号,客户姓名
),
所有客户 AS(
    SELECT 客户编号,客户姓名
        FROM CustomerInfo
)
SELECT c.客户编号,c.客户姓名,
    CASE
        WHEN t.最长连续天数 IS NULL THEN
            CASE
                WHEN EXISTS(
                    SELECT 1
                        FROM OrderList o
                        WHERE o.客户编号 = c.客户编号
                ) THEN '未连续下单'
                ELSE '未购物'
            END
```

```
    WHEN t.最长连续天数 = 1 THEN '未连续下单'
    ELSE CAST(t.最长连续天数 AS VARCHAR)
END AS 最长连续下单的天数
FROM 所有客户 c LEFT JOIN 最长连续天数 t ON c.客户编号 = t.客户编号
```

- 语句执行结果如下：

结果 消息

	客户编号	客户姓名	最长连续下单的天数
1	100001	张小林	4
2	100002	李红红	2
3	100003	王晓美	未购物
4	100004	赵明	未连续下单
5	100005	张帆一	未连续下单
6	100006	王芳芳	未连续下单
7	100007	赵明	未购物
8	100008	张小林	未购物
9	100009	张萍萍	未购物
10	100010	赵强	未购物
11	100011	张小林	未购物
12	100012	张小林	未购物
13	100013	李远	未购物
14	100014	王鹏	未购物
15	100015	王平平	未购物
16	100016	朱晓红	未购物

四、实验总结

此次实验我对子查询和综合查询更加了解，在实验过程中，我也遇到了许多问题，比如公共子查询如何实现，窗口函数用法等等，例如第 78 题中，为找出“最长连续下单天数”，需要用 ROW_NUMBER() 和日期偏移构造“分组键”，对我来说是一次高阶挑战，但也极大提升了我对窗口函数的理解。在学习了一些资料后，我初步掌握了这些函数用法，总而言之这次实验全面锻炼了我对 SQL 语言的掌握和实际数据分析能力。