# 数据库系统课程实验报告

| | |
|---|---|
| 实验名称： | 数据基本查询 |
| 实验日期： | 2022/4/7 |
| 实验地点： | 4 号楼 |
| 提交日期： | 2022/4/8 |

| | |
|---|---|
| 学号： | 22920202202879 |
| 姓名： | 陈奕培 |
| 专业年级： | 软工 2020 级 |
| 学年学期： | 2021-2022 学年第二学期 |

## 1.实验目的

- 熟练掌握 openGauss 单表查询的语法结构及其使用方法
- 掌握设计正确查询语句以实现查询要求的方法

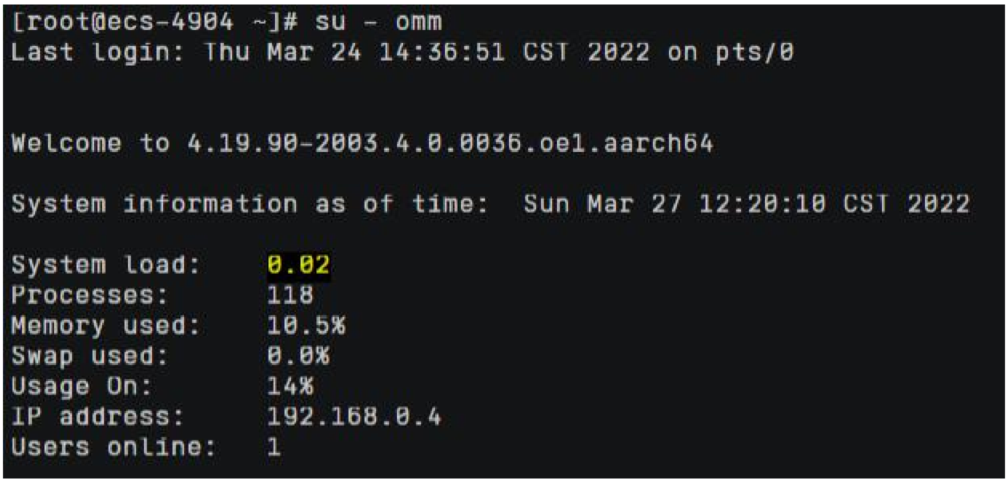
  普通单表查询、模糊查询、聚集函数、分组统计和排序
- 掌握 Group by 的使用
- 正确区分元组过滤条件（WHERE 子句）和分组过滤条件（HAVING 短语）的异同
- 掌握 Order by 的使用
- 掌握使用 DISTINCT 实现查询结果的去重方法
- 掌握空值 NULL 的使用方法
- 掌握查询过程中别名的使用方法
- 掌握自身连接的使用方法

## 2.实验内容和步骤

该部分的写作格式示例如下:

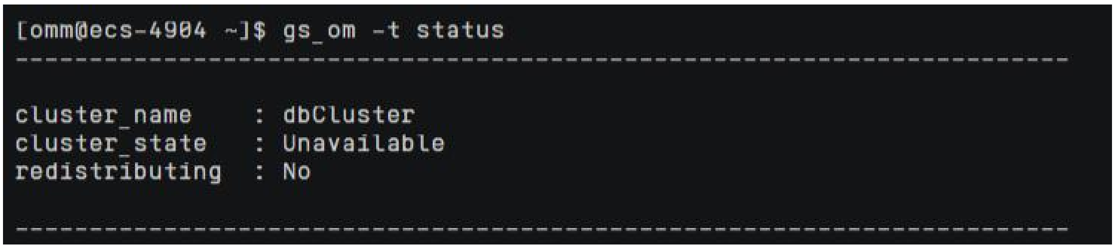
（1）连接到数据库

1.在数据库主节点服务器上，切换至 omm 操作系统用户环境。



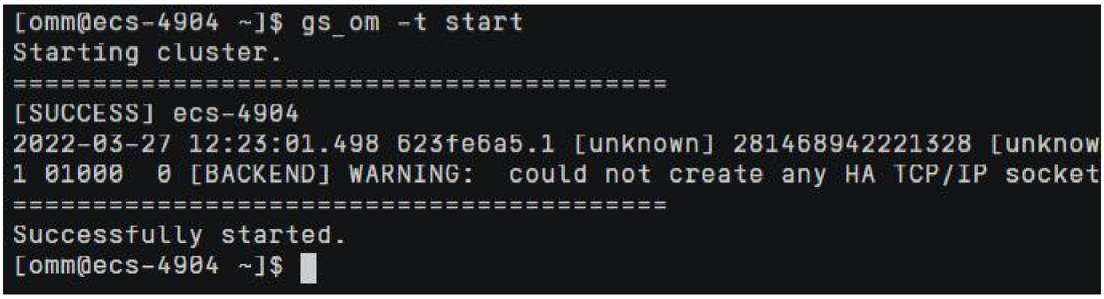
```
[root@ecs-4904 ~]# su - omm
Last login: Thu Mar 24 14:36:51 CST 2022 on pts/0


Welcome to 4.19.90-2003.4.0.0036.oe1.aarch64

System information as of time:  Sun Mar 27 12:20:10 CST 2022

System load:     0.02
Processes:       118
Memory used:     10.5%
Swap used:       0.0%
Usage On:        14%
IP address:      192.168.0.4
Users online:    1
```

2.查看服务是否启动。



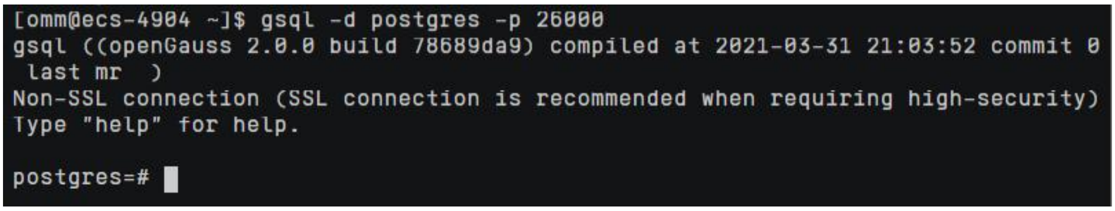
```
[omm@ecs-4904 ~]$ gs_om -t status
-----------------------------------------------------------------------

cluster_name     : dbCluster
cluster_state    : Unavailable
redistributing   : No

-----------------------------------------------------------------------
```

3.启动服务。



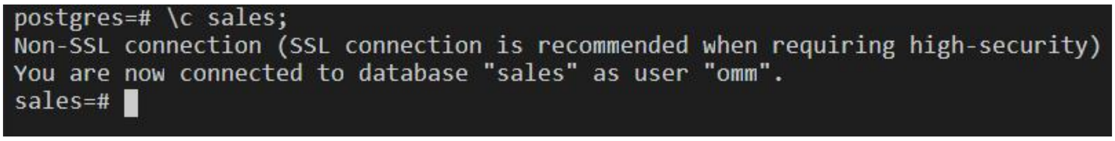
```
[omm@ecs-4904 ~]$ gs_om -t start
Starting cluster.
=========================================
[SUCCESS] ecs-4904
2022-03-27 12:23:01.498 623fe6a5.1 [unknown] 281468942221328 [unknow
1 01000  0 [BACKEND] WARNING:  could not create any HA TCP/IP socket
=========================================
Successfully started.
[omm@ecs-4904 ~]$ 
```

4.连接到数据库。



```
[omm@ecs-4904 ~]$ gsql -d postgres -p 26000
gsql ((openGauss 2.0.0 build 78689da9) compiled at 2021-03-31 21:03:52 commit 0
 last mr  )
Non-SSL connection (SSL connection is recommended when requiring high-security)
Type "help" for help.

postgres=# 
```

5.进入 sales 数据库

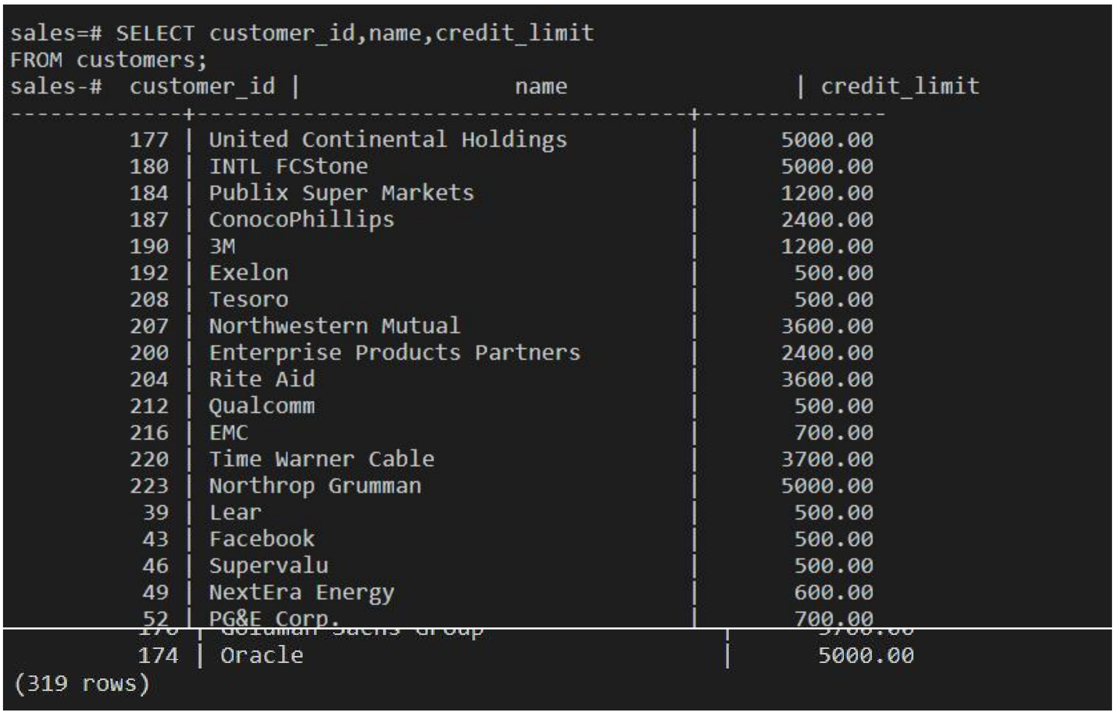


```
postgres=# \c sales;
Non-SSL connection (SSL connection is recommended when requiring high-security)
You are now connected to database "sales" as user "omm".
sales=# 
```

（2）查询

1. 查询顾客表中的顾客号、顾客名和信用卡额度

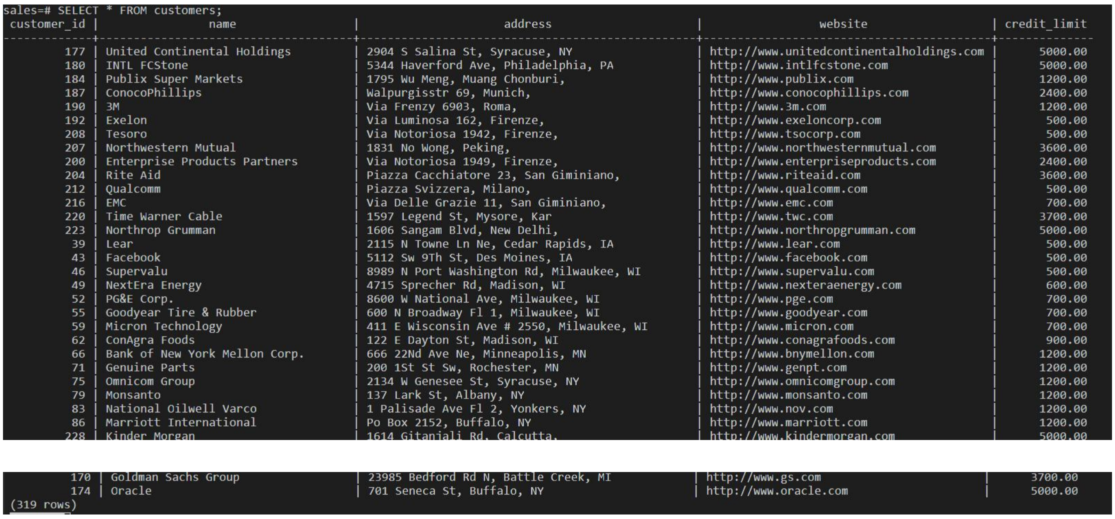
```sql
SELECT customer_id,name,credit_limit
FROM customers;
```



```
sales=# SELECT customer_id,name,credit_limit
FROM customers;
sales-#  customer_id |                  name                  | credit_limit
-------------+----------------------------------------+--------------
         177 | United Continental Holdings            |      5000.00
         180 | INTL FCStone                           |      5000.00
         184 | Publix Super Markets                   |      1200.00
         187 | ConocoPhillips                         |      2400.00
         190 | 3M                                     |      1200.00
         192 | Exelon                                 |       500.00
         208 | Tesoro                                 |       500.00
         207 | Northwestern Mutual                    |      3600.00
         200 | Enterprise Products Partners           |      2400.00
         204 | Rite Aid                               |      3600.00
         212 | Qualcomm                               |       500.00
         216 | EMC                                    |       700.00
         220 | Time Warner Cable                      |      3700.00
         223 | Northrop Grumman                       |      5000.00
          39 | Lear                                   |       500.00
          43 | Facebook                               |       500.00
          46 | Supervalu                              |       500.00
          49 | NextEra Energy                         |       600.00
          52 | PG&E Corp.                             |       700.00
         170 | Goldman Sachs Group                    |      3700.00
         174 | Oracle                                 |      5000.00
(319 rows)
```

## 2. 查询顾客的所有信息

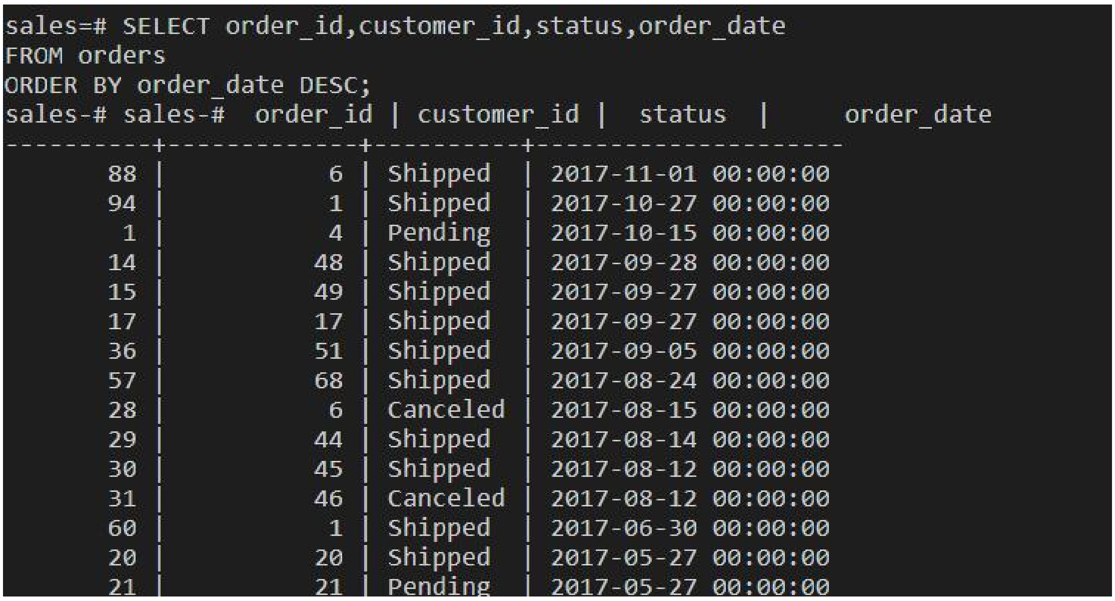
```
SELECT * FROM customers;
```



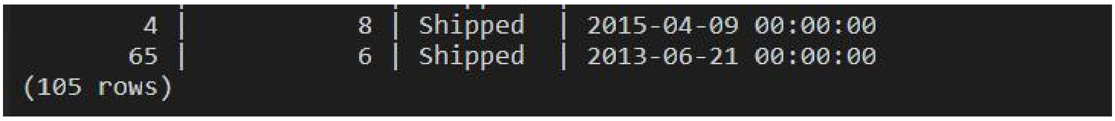
```
sales=# SELECT * FROM customers;
 customer_id |                  name                  |                 address                  |                  website                   | credit_limit
-------------+----------------------------------------+------------------------------------------+--------------------------------------------+--------------
         177 | United Continental Holdings            | 2904 S Salina St, Syracuse, NY           | http://www.unitedcontinentalholdings.com   |      5000.00
         180 | INTL FCStone                           | 5344 Haverford Ave, Philadelphia, PA     | http://www.intlfcstone.com                 |      5000.00
         184 | Publix Super Markets                   | 1795 Wu Meng, Muang Chonburi,            | http://www.publix.com                      |      1200.00
         187 | ConocoPhillips                         | Walpurgisstr 69, Munich,                 | http://www.conocophillips.com              |      2400.00
         190 | 3M                                     | Via Frenzy 6903, Roma,                   | http://www.3m.com                          |      1200.00
         192 | Exelon                                 | Via Luminosa 162, Firenze,               | http://www.exeloncorp.com                  |       500.00
         208 | Tesoro                                 | Via Notoriosa 1942, Firenze,             | http://www.tsocorp.com                     |       500.00
         207 | Northwestern Mutual                    | 1831 No Wong, Peking,                    | http://www.northwesternmutual.com          |      3600.00
         200 | Enterprise Products Partners           | Via Notoriosa 1949, Firenze,             | http://www.enterpriseproducts.com          |      2400.00
         204 | Rite Aid                               | Piazza Cacchiatore 23, San Giminiano,    | http://www.riteaid.com                     |      3600.00
         212 | Qualcomm                               | Piazza Svizzera, Milano,                 | http://www.qualcomm.com                    |       500.00
         216 | EMC                                    | Via Delle Grazie 11, San Giminiano,      | http://www.emc.com                         |       700.00
         220 | Time Warner Cable                      | 1597 Legend St, Mysore, Kar              | http://www.twc.com                         |      3700.00
         223 | Northrop Grumman                       | 1606 Sangam Blvd, New Delhi,             | http://www.northropgrumman.com             |      5000.00
          39 | Lear                                   | 2115 N Towne Ln Ne, Cedar Rapids, IA     | http://www.lear.com                        |       500.00
          43 | Facebook                               | 5112 Sw 9Th St, Des Moines, IA           | http://www.facebook.com                    |       500.00
          46 | Supervalu                              | 8989 N Port Washington Rd, Milwaukee, WI | http://www.supervalu.com                   |       500.00
          49 | NextEra Energy                         | 4715 Sprecher Rd, Madison, WI            | http://www.nexteraenergy.com               |       600.00
          52 | PG&E Corp.                             | 8600 W National Ave, Milwaukee, WI       | http://www.pge.com                         |       700.00
          55 | Goodyear Tire & Rubber                 | 600 N Broadway Fl 1, Milwaukee, WI       | http://www.goodyear.com                    |       700.00
          59 | Micron Technology                      | 411 E Wisconsin Ave # 2550, Milwaukee, WI | http://www.micron.com                     |       700.00
          62 | ConAgra Foods                          | 122 E Dayton St, Madison, WI             | http://www.conagrafoods.com                |       900.00
          66 | Bank of New York Mellon Corp.          | 666 22Nd Ave Ne, Minneapolis, MN         | http://www.bnymellon.com                   |      1200.00
          71 | Genuine Parts                          | 200 1St St Sw, Rochester, MN             | http://www.genpt.com                       |      1200.00
          75 | Omnicom Group                          | 2134 W Genesee St, Syracuse, NY          | http://www.omnicomgroup.com                |      1200.00
          79 | Monsanto                               | 137 Lark St, Albany, NY                  | http://www.monsanto.com                    |      1200.00
          83 | National Oilwell Varco                 | 1 Palisade Ave Fl 2, Yonkers, NY         | http://www.nov.com                         |      1200.00
          86 | Marriott International                 | Po Box 2152, Buffalo, NY                 | http://www.marriott.com                    |      1200.00
         228 | Kinder Morgan                          | 1614 Gitanjali Rd, Calcutta,             | http://www.kindermorgan.com                |      5000.00

         170 | Goldman Sachs Group                    | 23985 Bedford Rd N, Battle Creek, MI     | http://www.gs.com                          |      3700.00
         174 | Oracle                                 | 701 Seneca St, Buffalo, NY               | http://www.oracle.com                      |      5000.00
(319 rows)
```

## 3. 查询订单表中的订单号，顾客号，状态，订单日期，并按订单日期降序显示结果

```
SELECT order_id,customer_id,status,order_date
FROM orders
ORDER BY order_date DESC;
```



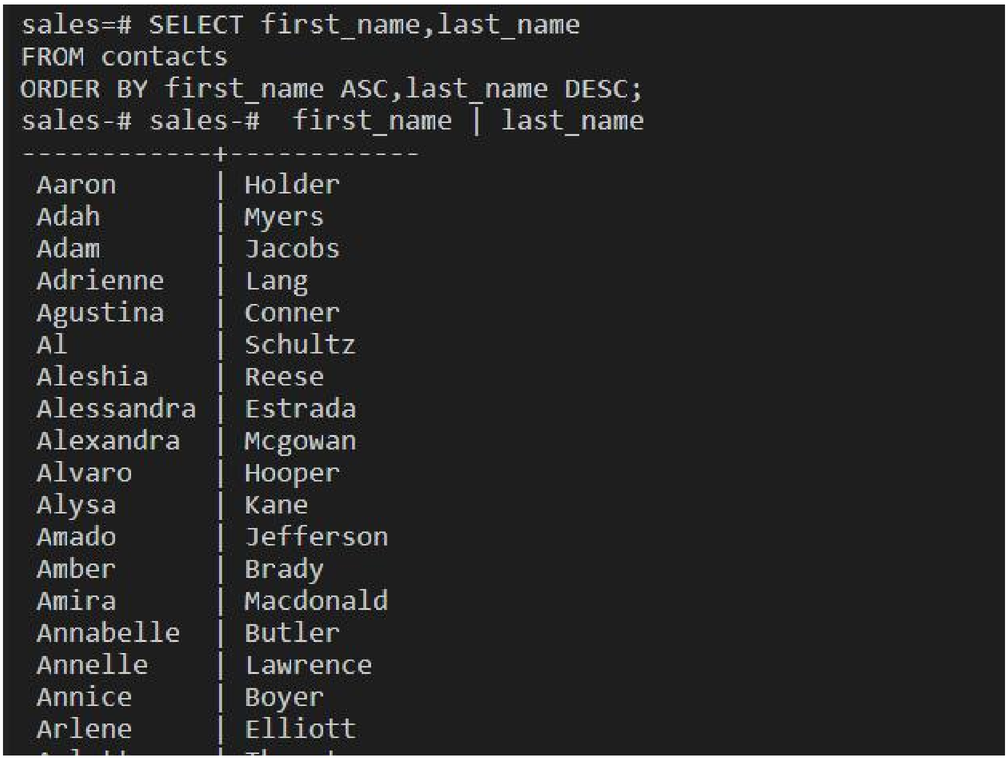
```
sales=# SELECT order_id,customer_id,status,order_date
FROM orders
ORDER BY order_date DESC;
sales-# sales-#  order_id | customer_id |  status  |     order_date
----------+-------------+----------+---------------------
       88 |           6 | Shipped  | 2017-11-01 00:00:00
       94 |           1 | Shipped  | 2017-10-27 00:00:00
        1 |           4 | Pending  | 2017-10-15 00:00:00
       14 |          48 | Shipped  | 2017-09-28 00:00:00
       15 |          49 | Shipped  | 2017-09-27 00:00:00
       17 |          17 | Shipped  | 2017-09-27 00:00:00
       36 |          51 | Shipped  | 2017-09-05 00:00:00
       57 |          68 | Shipped  | 2017-08-24 00:00:00
       28 |           6 | Canceled | 2017-08-15 00:00:00
       29 |          44 | Shipped  | 2017-08-14 00:00:00
       30 |          45 | Shipped  | 2017-08-12 00:00:00
       31 |          46 | Canceled | 2017-08-12 00:00:00
       60 |           1 | Shipped  | 2017-06-30 00:00:00
       20 |          20 | Shipped  | 2017-05-27 00:00:00
       21 |          21 | Pending  | 2017-05-27 00:00:00
```



```
        4 |           8 | Shipped  | 2015-04-09 00:00:00
       65 |           6 | Shipped  | 2013-06-21 00:00:00
(105 rows)
```

4. 查询联系表中的名和姓，并按名升序，姓降序显示

```
SELECT first_name,last_name
FROM contacts
ORDER BY first_name ASC,last_name DESC;
```



```
sales=# SELECT first_name,last_name
FROM contacts
ORDER BY first_name ASC,last_name DESC;
sales-# sales-#  first_name | last_name
------------+------------
 Aaron      | Holder
 Adah       | Myers
 Adam       | Jacobs
 Adrienne   | Lang
 Agustina   | Conner
 Al         | Schultz
 Aleshia    | Reese
 Alessandra | Estrada
 Alexandra  | Mcgowan
 Alvaro     | Hooper
 Alysa      | Kane
 Amado      | Jefferson
 Amber      | Brady
 Amira      | Macdonald
 Annabelle  | Butler
 Annelle    | Lawrence
 Annice     | Boyer
 Arlene     | Elliott
```

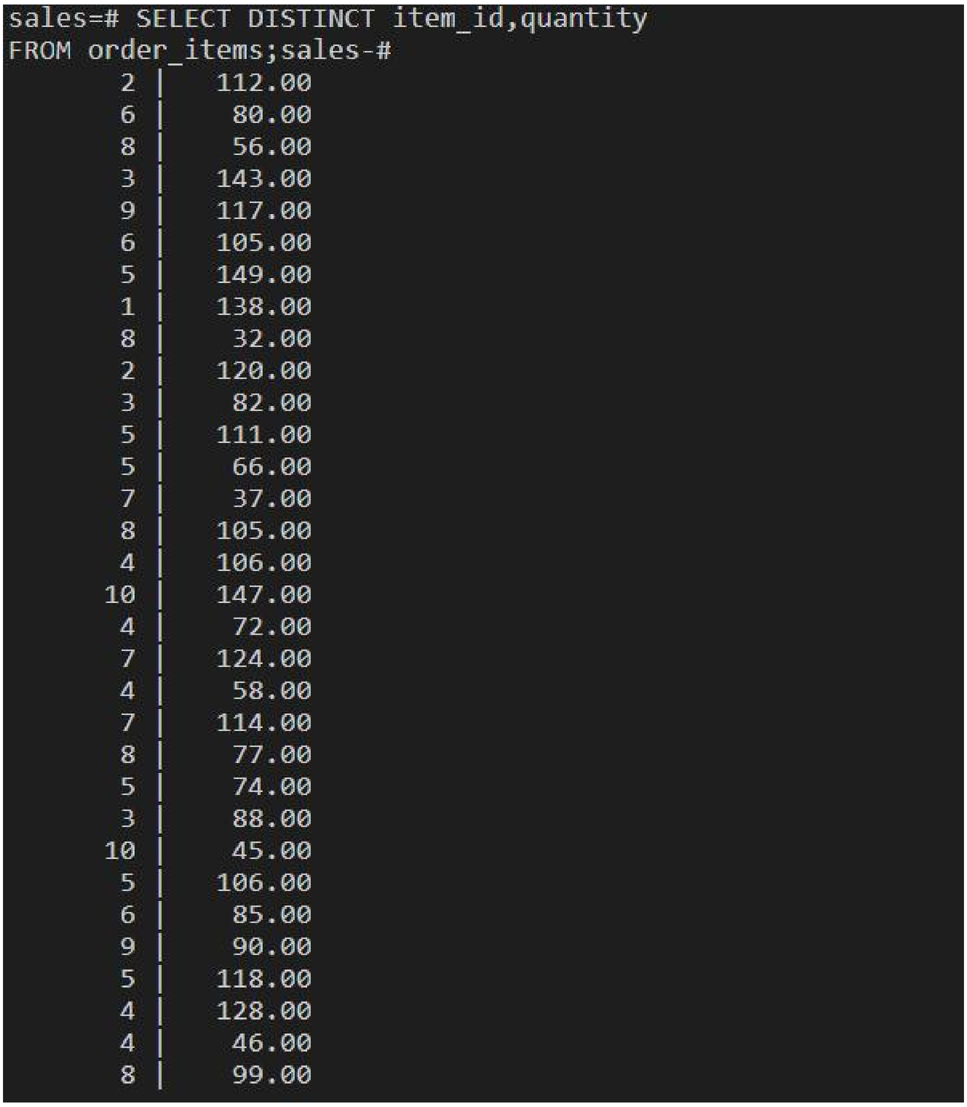
5. 执行以下语句并观察 state 列 NULL 值的显示位置，得出结论

```
SELECT country_id, city,state FROM locations ORDER BY city,state;
SELECT country_id, city,state FROM locations ORDER BY state ASC NULLS LAST;
SELECT country_id, city,state FROM locations ORDER BY state ASC NULLS
FIRST ;
```

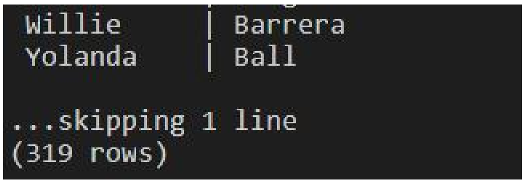
设置 NULLS FIRST 可以使空值排在前面，反之 LAST 排在后面

6. 查询订单细节表中的产品号和数量，查询结果应无重复元组

```
SELECT DISTINCT item_id,quantity
FROM order_items;
```



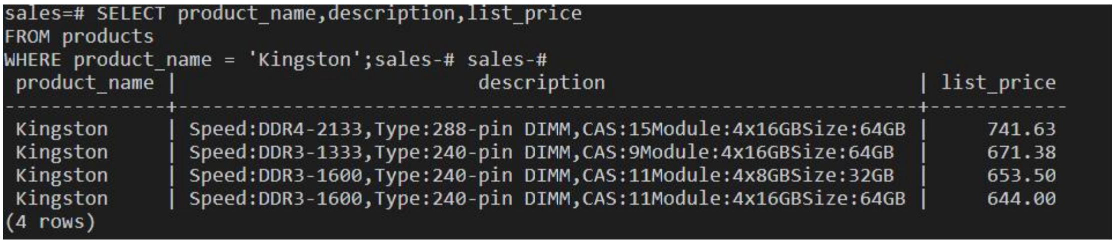
```
sales=# SELECT DISTINCT item_id,quantity
FROM order_items;sales-#
       2 |   112.00
       6 |    80.00
       8 |    56.00
       3 |   143.00
       9 |   117.00
       6 |   105.00
       5 |   149.00
       1 |   138.00
       8 |    32.00
       2 |   120.00
       3 |    82.00
       5 |   111.00
       5 |    66.00
       7 |    37.00
       8 |   105.00
       4 |   106.00
      10 |   147.00
       4 |    72.00
       7 |   124.00
       4 |    58.00
       7 |   114.00
       8 |    77.00
       5 |    74.00
       3 |    88.00
      10 |    45.00
       5 |   106.00
       6 |    85.00
       9 |    90.00
       5 |   118.00
       4 |   128.00
       4 |    46.00
       8 |    99.00
```



```
 Willie      | Barrera
 Yolanda     | Ball

...skipping 1 line
(319 rows)
```

7. 查询产品表中的产品名为‘Kingston’的产品名，产品描述和价格

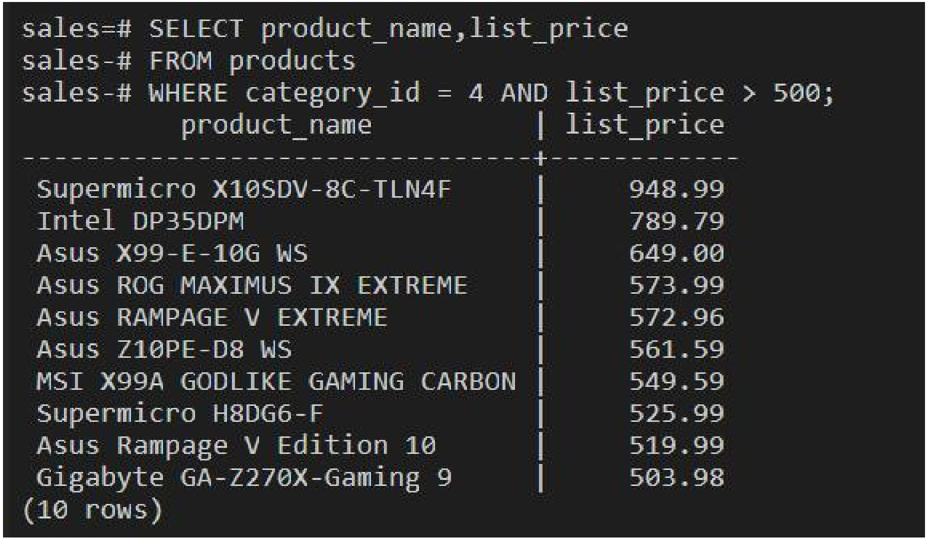
```
SELECT product_name,description,list_price
FROM products
WHERE product_name = 'Kingston';
```



```
sales=# SELECT product_name,description,list_price
FROM products
WHERE product_name = 'Kingston';sales-# sales-#
 product_name |                          description                           | list_price
--------------+----------------------------------------------------------------+------------
 Kingston     | Speed:DDR4-2133,Type:288-pin DIMM,CAS:15Module:4x16GBSize:64GB |     741.63
 Kingston     | Speed:DDR3-1333,Type:240-pin DIMM,CAS:9Module:4x16GBSize:64GB  |     671.38
 Kingston     | Speed:DDR3-1600,Type:240-pin DIMM,CAS:11Module:4x8GBSize:32GB  |     653.50
 Kingston     | Speed:DDR3-1600,Type:240-pin DIMM,CAS:11Module:4x16GBSize:64GB |     644.00
(4 rows)
```

8. 查询产品表中所有价格大于 500 且 category_id 为 4 的产品名和价格

```
SELECT product_name,list_price
FROM products
WHERE category_id = 4 AND list_price > 500;
```

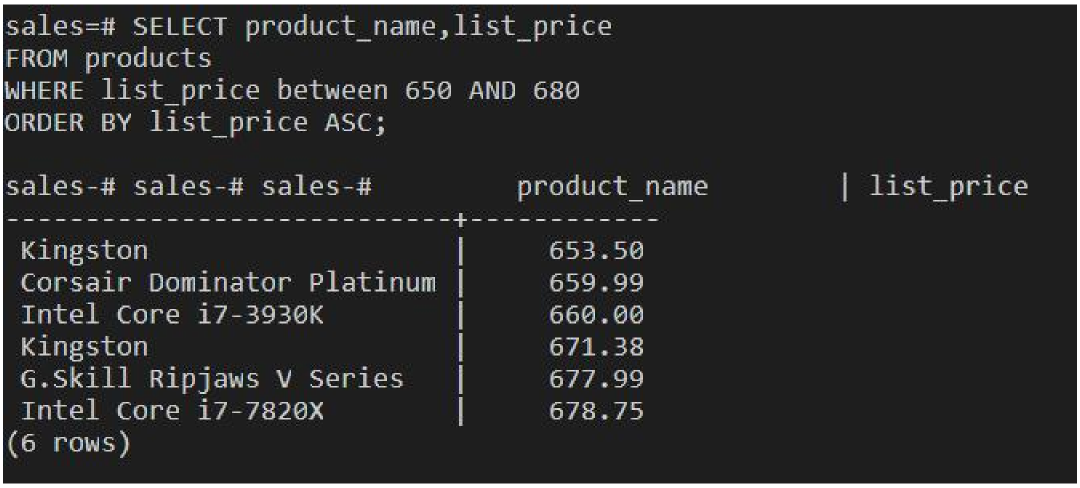


```
sales=# SELECT product_name,list_price
sales-# FROM products
sales-# WHERE category_id = 4 AND list_price > 500;
          product_name          | list_price
--------------------------------+------------
 Supermicro X10SDV-8C-TLN4F     |     948.99
 Intel DP35DPM                  |     789.79
 Asus X99-E-10G WS              |     649.00
 Asus ROG MAXIMUS IX EXTREME    |     573.99
 Asus RAMPAGE V EXTREME         |     572.96
 Asus Z10PE-D8 WS               |     561.59
 MSI X99A GODLIKE GAMING CARBON |     549.59
 Supermicro H8DG6-F             |     525.99
 Asus Rampage V Edition 10      |     519.99
 Gigabyte GA-Z270X-Gaming 9     |     503.98
(10 rows)
```

9. 查询产品表中所有价格在 650 和 680 之间的产品名和价格并按价格升序显示结果

```
SELECT product_name,list_price
FROM products
```

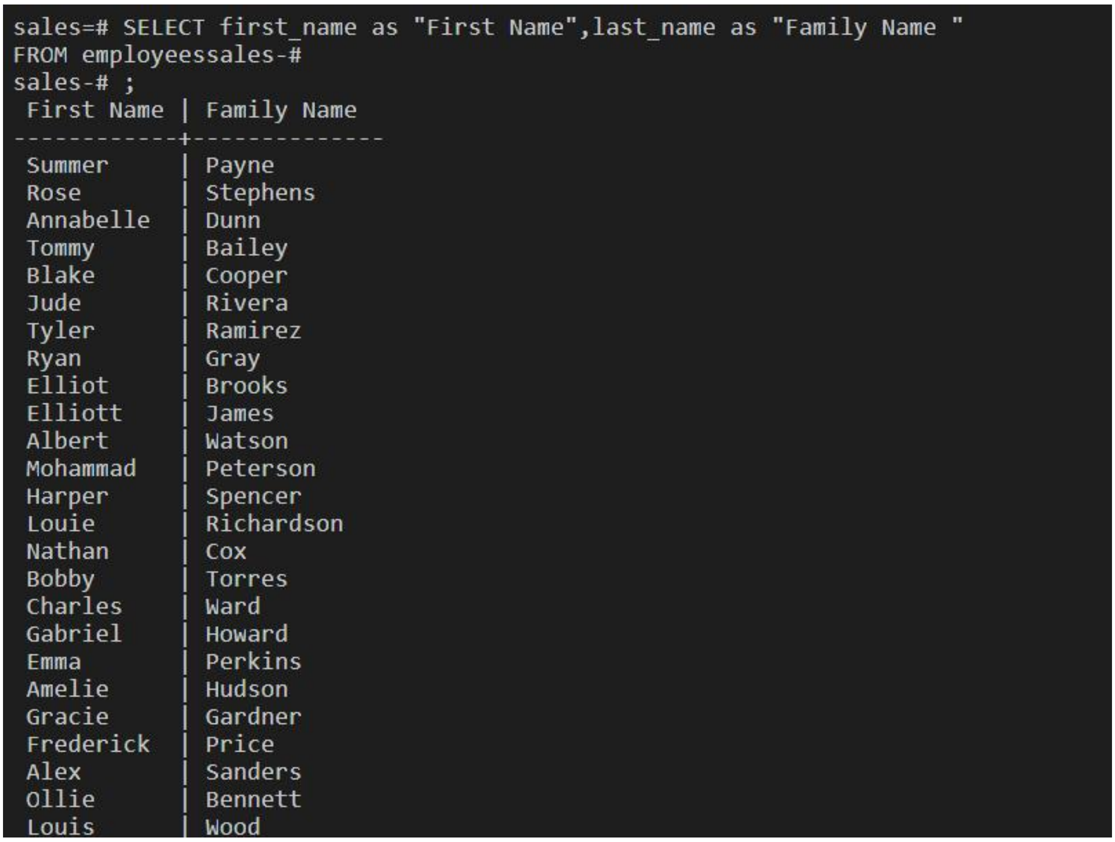
```
WHERE list_price BETWEEN 650 AND 680
ORDER BY list_price ASC;
```



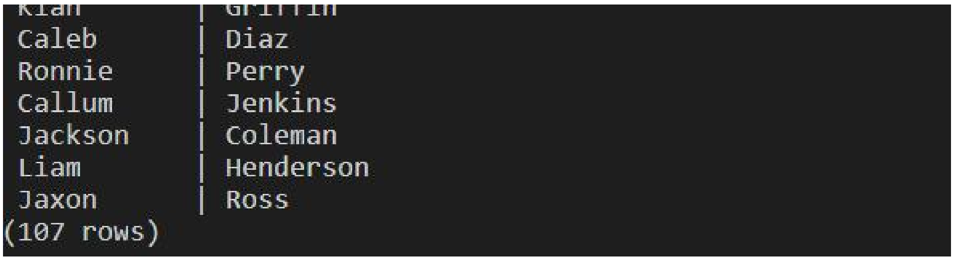
```
sales=# SELECT product_name,list_price
FROM products
WHERE list_price between 650 AND 680
ORDER BY list_price ASC;

sales-# sales-# sales-#         product_name         | list_price
-----------------------------+------------
 Kingston                    |     653.50
 Corsair Dominator Platinum |     659.99
 Intel Core i7-3930K         |     660.00
 Kingston                    |     671.38
 G.Skill Ripjaws V Series    |     677.99
 Intel Core i7-7820X         |     678.75
(6 rows)
```

10. 查询雇员表中的名和姓，名和姓的字段分别显示为"First Name"和"Family Name"

```
SELECT first_name AS "First Name",last_name AS "Family Name "
FROM employees
```



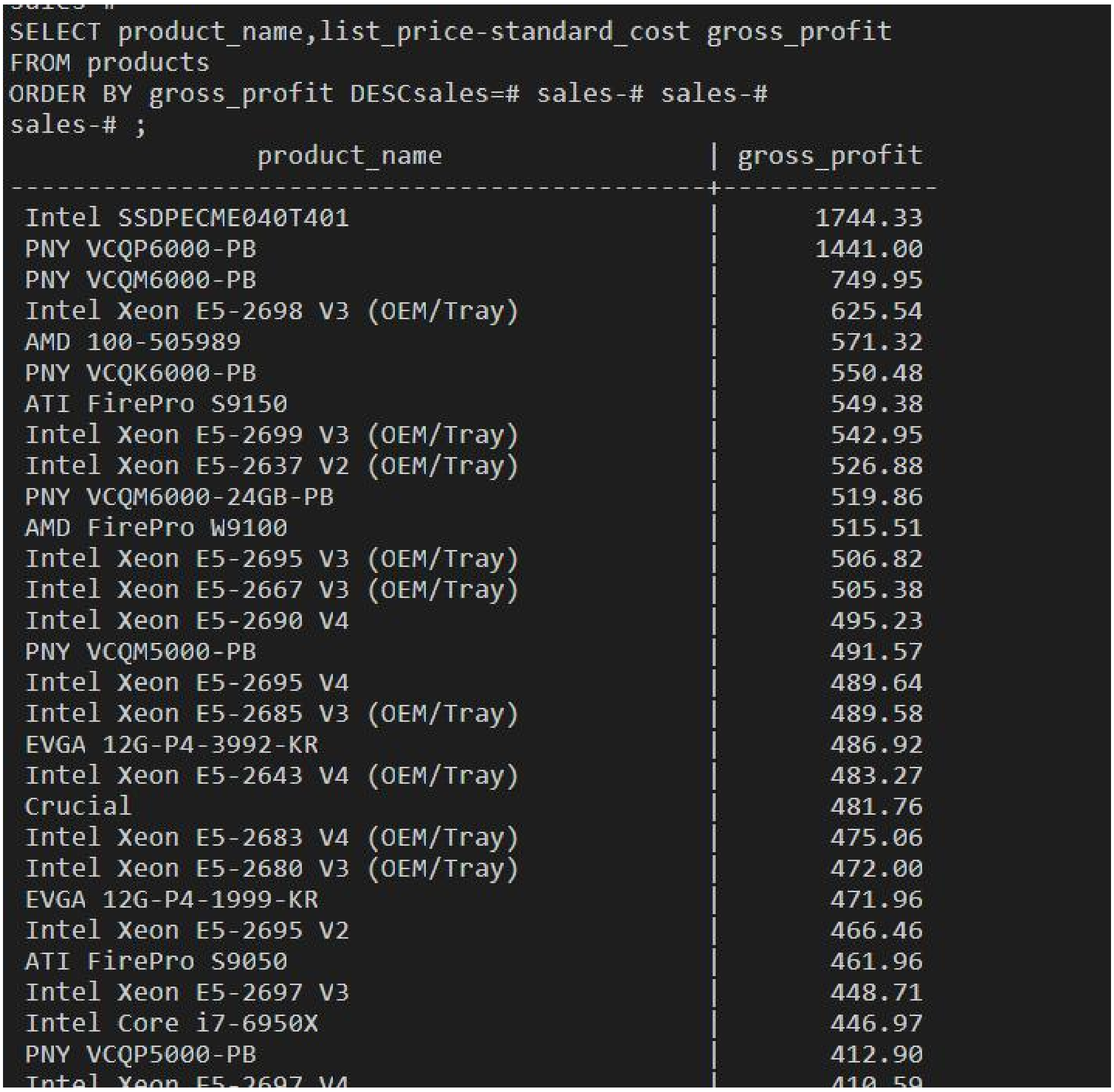
```
sales=# SELECT first_name as "First Name",last_name as "Family Name "
FROM employeessales-#
sales-# ;
 First Name | Family Name
------------+-------------
 Summer     | Payne
 Rose       | Stephens
 Annabelle  | Dunn
 Tommy      | Bailey
 Blake      | Cooper
 Jude       | Rivera
 Tyler      | Ramirez
 Ryan       | Gray
 Elliot     | Brooks
 Elliott    | James
 Albert     | Watson
 Mohammad   | Peterson
 Harper     | Spencer
 Louie      | Richardson
 Nathan     | Cox
 Bobby      | Torres
 Charles    | Ward
 Gabriel    | Howard
 Emma       | Perkins
 Amelie     | Hudson
 Gracie     | Gardner
 Frederick  | Price
 Alex       | Sanders
 Ollie      | Bennett
 Louis      | Wood
```



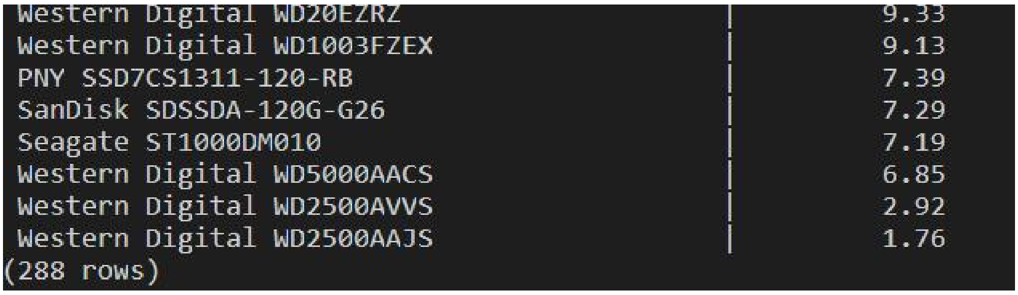
```
 Kian         | Griffin
 Caleb        | Diaz
 Ronnie       | Perry
 Callum       | Jenkins
 Jackson      | Coleman
 Liam         | Henderson
 Jaxon        | Ross
(107 rows)
```

11. 查询产品表中的产品名及毛利，并按毛利结果降序显示，毛利名为 gross_profit

```
SELECT product_name,list_price-standard_cost gross_profit
FROM products
ORDER BY gross_profit DESC;
```



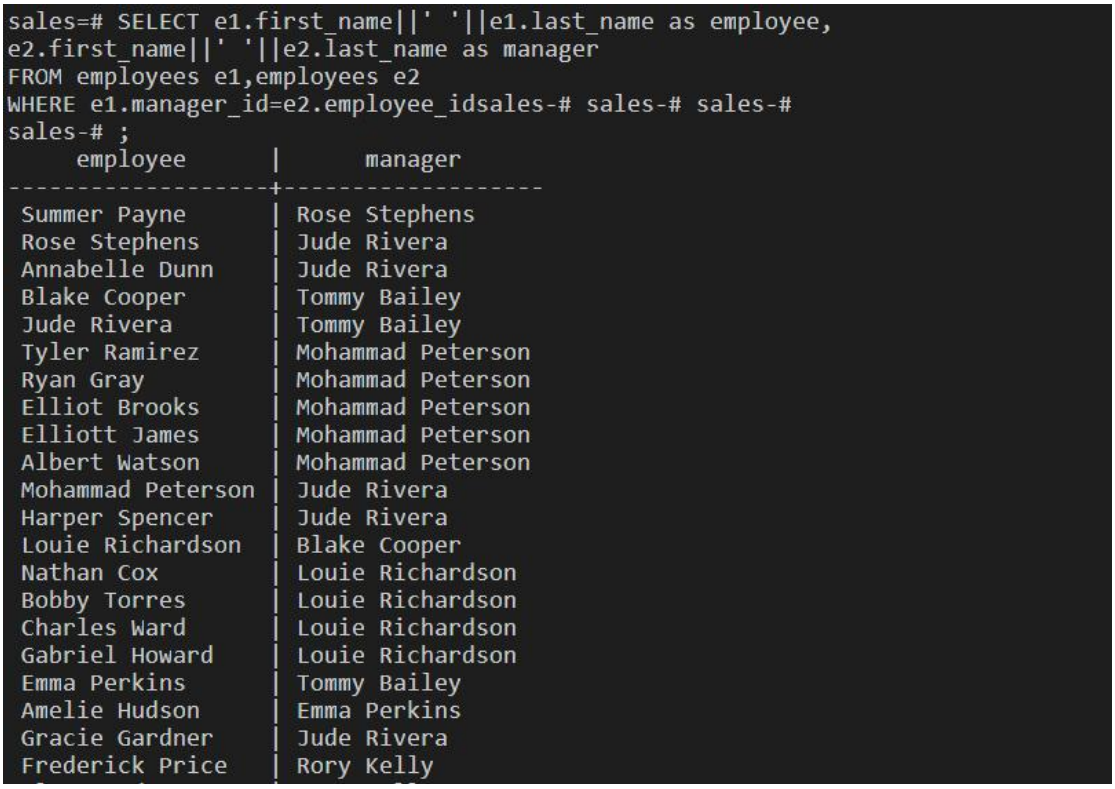
```
sales-#
SELECT product_name,list_price-standard_cost gross_profit
FROM products
ORDER BY gross_profit DESCsales=# sales-# sales-#
sales-# ;
                product_name                 | gross_profit
---------------------------------------------+--------------
 Intel SSDPECME040T401                       |      1744.33
 PNY VCQP6000-PB                             |      1441.00
 PNY VCQM6000-PB                             |       749.95
 Intel Xeon E5-2698 V3 (OEM/Tray)            |       625.54
 AMD 100-505989                              |       571.32
 PNY VCQK6000-PB                             |       550.48
 ATI FirePro S9150                           |       549.38
 Intel Xeon E5-2699 V3 (OEM/Tray)            |       542.95
 Intel Xeon E5-2637 V2 (OEM/Tray)            |       526.88
 PNY VCQM6000-24GB-PB                        |       519.86
 AMD FirePro W9100                           |       515.51
 Intel Xeon E5-2695 V3 (OEM/Tray)            |       506.82
 Intel Xeon E5-2667 V3 (OEM/Tray)            |       505.38
 Intel Xeon E5-2690 V4                       |       495.23
 PNY VCQM5000-PB                             |       491.57
 Intel Xeon E5-2695 V4                       |       489.64
 Intel Xeon E5-2685 V3 (OEM/Tray)            |       489.58
 EVGA 12G-P4-3992-KR                         |       486.92
 Intel Xeon E5-2643 V4 (OEM/Tray)            |       483.27
 Crucial                                     |       481.76
 Intel Xeon E5-2683 V4 (OEM/Tray)            |       475.06
 Intel Xeon E5-2680 V3 (OEM/Tray)            |       472.00
 EVGA 12G-P4-1999-KR                         |       471.96
 Intel Xeon E5-2695 V2                       |       466.46
 ATI FirePro S9050                           |       461.96
 Intel Xeon E5-2697 V3                       |       448.71
 Intel Core i7-6950X                         |       446.97
 PNY VCQP5000-PB                             |       412.90
 Intel Xeon E5-2697 V4                       |       410.59
```



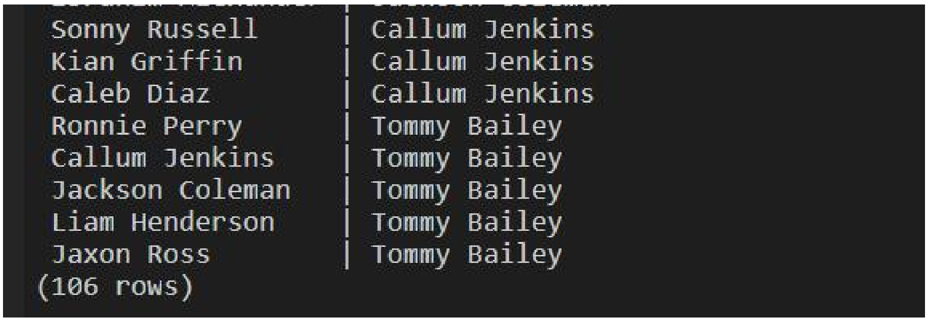
```
 Western Digital WD20EZRZ                    |        9.33
 Western Digital WD1003FZEX                  |        9.13
 PNY SSD7CS1311-120-RB                       |        7.39
 SanDisk SDSSDA-120G-G26                     |        7.29
 Seagate ST1000DM010                         |        7.19
 Western Digital WD5000AACS                  |        6.85
 Western Digital WD2500AVVS                  |        2.92
 Western Digital WD2500AAJS                  |        1.76
(288 rows)
```

12. 查询雇员表中每个雇员对应的经理名，要求第一列字段名为 employee，第二列字段名为 manager，无其他字段

```
SELECT e1.first_name||' '||e1.last_name AS employee,
e2.first_name||' '||e2.last_name AS manager
FROM employees e1,employees e2
WHERE e1.manager_id=e2.employee_id
```



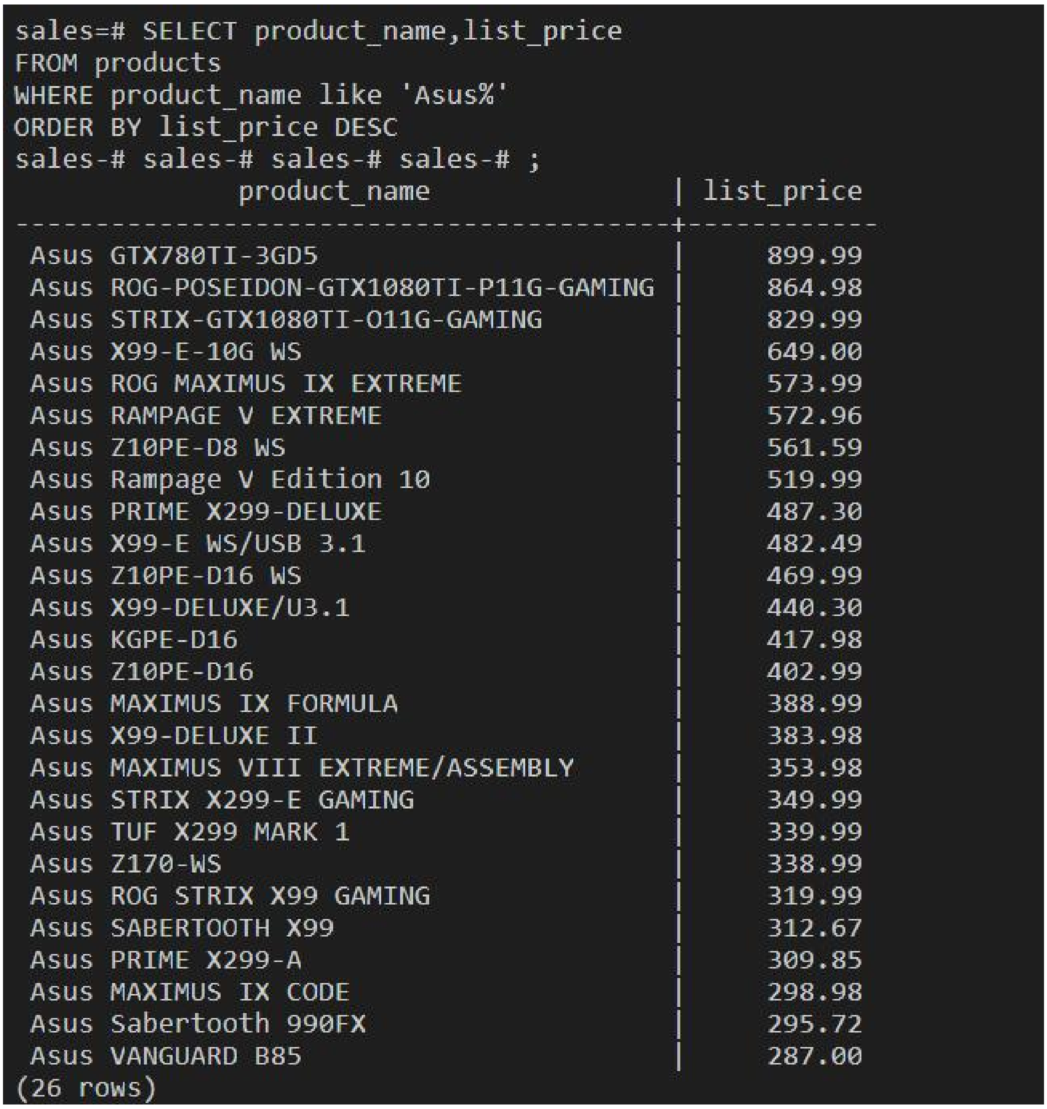
```
sales=# SELECT e1.first_name||' '||e1.last_name as employee,
e2.first_name||' '||e2.last_name as manager
FROM employees e1,employees e2
WHERE e1.manager_id=e2.employee_idsales-# sales-# sales-#
sales-# ;
     employee      |      manager
-------------------+-------------------
 Summer Payne      | Rose Stephens
 Rose Stephens     | Jude Rivera
 Annabelle Dunn    | Jude Rivera
 Blake Cooper      | Tommy Bailey
 Jude Rivera       | Tommy Bailey
 Tyler Ramirez     | Mohammad Peterson
 Ryan Gray         | Mohammad Peterson
 Elliot Brooks     | Mohammad Peterson
 Elliott James     | Mohammad Peterson
 Albert Watson     | Mohammad Peterson
 Mohammad Peterson | Jude Rivera
 Harper Spencer    | Jude Rivera
 Louie Richardson  | Blake Cooper
 Nathan Cox        | Louie Richardson
 Bobby Torres      | Louie Richardson
 Charles Ward      | Louie Richardson
 Gabriel Howard    | Louie Richardson
 Emma Perkins      | Tommy Bailey
 Amelie Hudson     | Emma Perkins
 Gracie Gardner    | Jude Rivera
 Frederick Price   | Rory Kelly
```



```
 Sonny Russell     | Callum Jenkins
 Kian Griffin      | Callum Jenkins
 Caleb Diaz        | Callum Jenkins
 Ronnie Perry      | Tommy Bailey
 Callum Jenkins    | Tommy Bailey
 Jackson Coleman   | Tommy Bailey
 Liam Henderson    | Tommy Bailey
 Jaxon Ross        | Tommy Bailey
(106 rows)
```

13. 查询产品表中所有以 Asus 开头的产品名和价格，并以价格降序显示

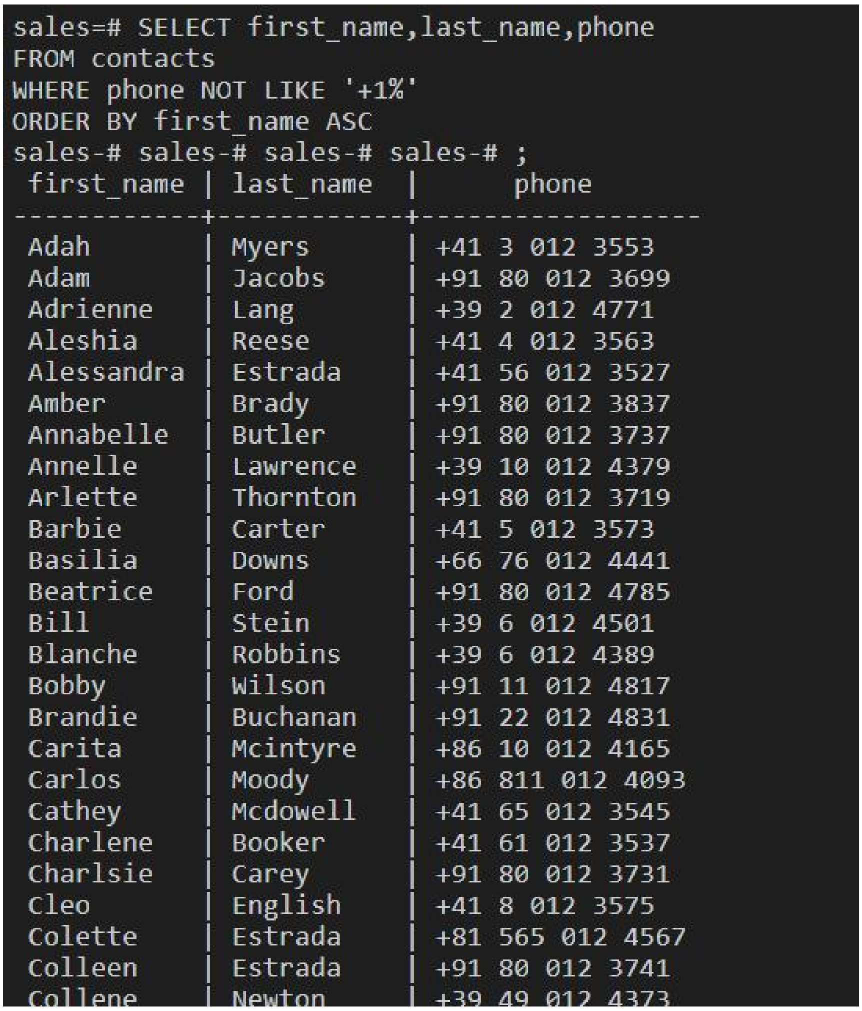
```
SELECT product_name,list_price
FROM products
WHERE product_name LIKE 'Asus%'
ORDER BY list_price DESC;
```



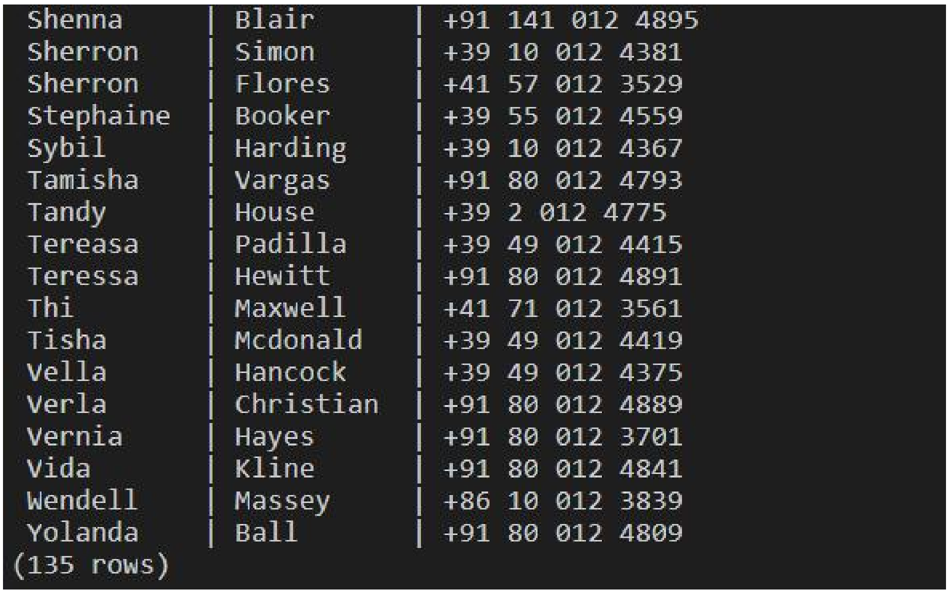
```
sales=# SELECT product_name,list_price
FROM products
WHERE product_name like 'Asus%'
ORDER BY list_price DESC
sales-# sales-# sales-# sales-# ;
              product_name                 | list_price
-------------------------------------------+------------
 Asus GTX780TI-3GD5                        |     899.99
 Asus ROG-POSEIDON-GTX1080TI-P11G-GAMING   |     864.98
 Asus STRIX-GTX1080TI-O11G-GAMING          |     829.99
 Asus X99-E-10G WS                         |     649.00
 Asus ROG MAXIMUS IX EXTREME               |     573.99
 Asus RAMPAGE V EXTREME                    |     572.96
 Asus Z10PE-D8 WS                          |     561.59
 Asus Rampage V Edition 10                 |     519.99
 Asus PRIME X299-DELUXE                    |     487.30
 Asus X99-E WS/USB 3.1                     |     482.49
 Asus Z10PE-D16 WS                         |     469.99
 Asus X99-DELUXE/U3.1                      |     440.30
 Asus KGPE-D16                             |     417.98
 Asus Z10PE-D16                            |     402.99
 Asus MAXIMUS IX FORMULA                   |     388.99
 Asus X99-DELUXE II                        |     383.98
 Asus MAXIMUS VIII EXTREME/ASSEMBLY        |     353.98
 Asus STRIX X299-E GAMING                  |     349.99
 Asus TUF X299 MARK 1                      |     339.99
 Asus Z170-WS                              |     338.99
 Asus ROG STRIX X99 GAMING                 |     319.99
 Asus SABERTOOTH X99                       |     312.67
 Asus PRIME X299-A                         |     309.85
 Asus MAXIMUS IX CODE                      |     298.98
 Asus Sabertooth 990FX                     |     295.72
 Asus VANGUARD B85                         |     287.00
(26 rows)
```

14. 查询联系表中电话号码不是以‘+1’开头的名、姓和电话号码，并以名升序显示

```
SELECT first_name,last_name,phone
FROM contacts
WHERE phone NOT LIKE '+1%'
ORDER BY first_name ASC;
```



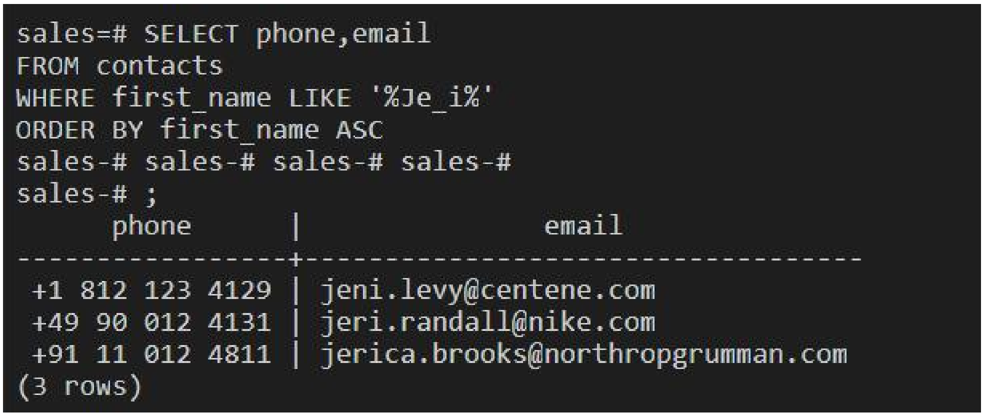
```
sales=# SELECT first_name,last_name,phone
FROM contacts
WHERE phone NOT LIKE '+1%'
ORDER BY first_name ASC
sales-# sales-# sales-# sales-# ;
 first_name | last_name  |      phone
------------+------------+------------------
 Adah       | Myers      | +41 3 012 3553
 Adam       | Jacobs     | +91 80 012 3699
 Adrienne   | Lang       | +39 2 012 4771
 Aleshia    | Reese      | +41 4 012 3563
 Alessandra | Estrada    | +41 56 012 3527
 Amber      | Brady      | +91 80 012 3837
 Annabelle  | Butler     | +91 80 012 3737
 Annelle    | Lawrence   | +39 10 012 4379
 Arlette    | Thornton   | +91 80 012 3719
 Barbie     | Carter     | +41 5 012 3573
 Basilia    | Downs      | +66 76 012 4441
 Beatrice   | Ford       | +91 80 012 4785
 Bill       | Stein      | +39 6 012 4501
 Blanche    | Robbins    | +39 6 012 4389
 Bobby      | Wilson     | +91 11 012 4817
 Brandie    | Buchanan   | +91 22 012 4831
 Carita     | Mcintyre   | +86 10 012 4165
 Carlos     | Moody      | +86 811 012 4093
 Cathey     | Mcdowell   | +41 65 012 3545
 Charlene   | Booker     | +41 61 012 3537
 Charlsie   | Carey      | +91 80 012 3731
 Cleo       | English    | +41 8 012 3575
 Colette    | Estrada    | +81 565 012 4567
 Colleen    | Estrada    | +91 80 012 3741
 Collene    | Newton     | +39 49 012 4373
```



```
 Shenna     | Blair      | +91 141 012 4895
 Sherron    | Simon      | +39 10 012 4381
 Sherron    | Flores     | +41 57 012 3529
 Stephaine  | Booker     | +39 55 012 4559
 Sybil      | Harding    | +39 10 012 4367
 Tamisha    | Vargas     | +91 80 012 4793
 Tandy      | House      | +39 2 012 4775
 Tereasa    | Padilla    | +39 49 012 4415
 Teressa    | Hewitt     | +91 80 012 4891
 Thi        | Maxwell    | +41 71 012 3561
 Tisha      | Mcdonald   | +39 49 012 4419
 Vella      | Hancock    | +39 49 012 4375
 Verla      | Christian  | +91 80 012 4889
 Vernia     | Hayes      | +91 80 012 3701
 Vida       | Kline      | +91 80 012 4841
 Wendell    | Massey     | +86 10 012 3839
 Yolanda    | Ball       | +91 80 012 4809
(135 rows)
```

15. 查询联系表中的电话号码和电子邮件，要求名中包含‘Je_i’且以名升序显示

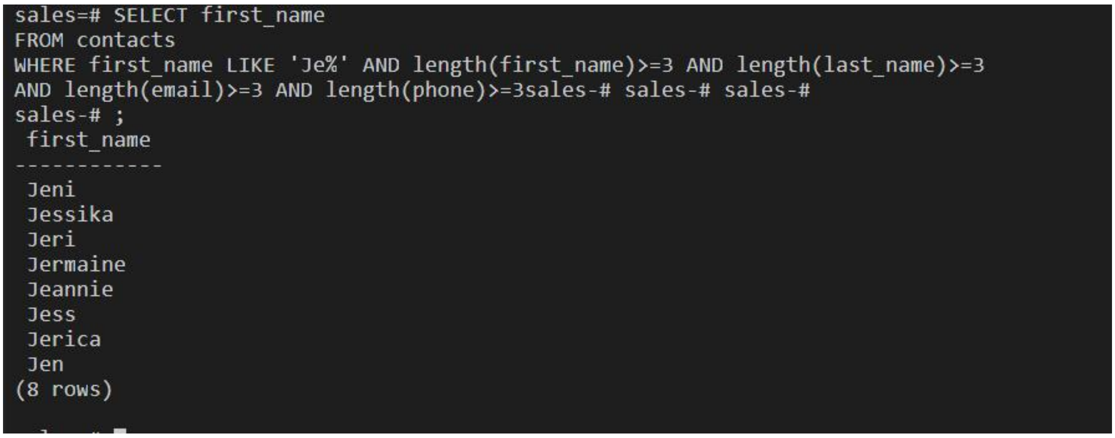
```sql
SELECT phone,email
FROM contacts
WHERE first_name LIKE '%Je_i%'
ORDER BY first_name ASC;
```



```
sales=# SELECT phone,email
FROM contacts
WHERE first_name LIKE '%Je_i%'
ORDER BY first_name ASC
sales-# sales-# sales-# sales-#
sales-# ;
      phone       |                email
-----------------+-----------------------------------
 +1 812 123 4129 | jeni.levy@centene.com
 +49 90 012 4131 | jeri.randall@nike.com
 +91 11 012 4811 | jerica.brooks@northropgrumman.com
(3 rows)
```

16. 查询联系表中所有以开头'Je'的名，且至少包含 3 个字符的名，姓，电子邮件和电话

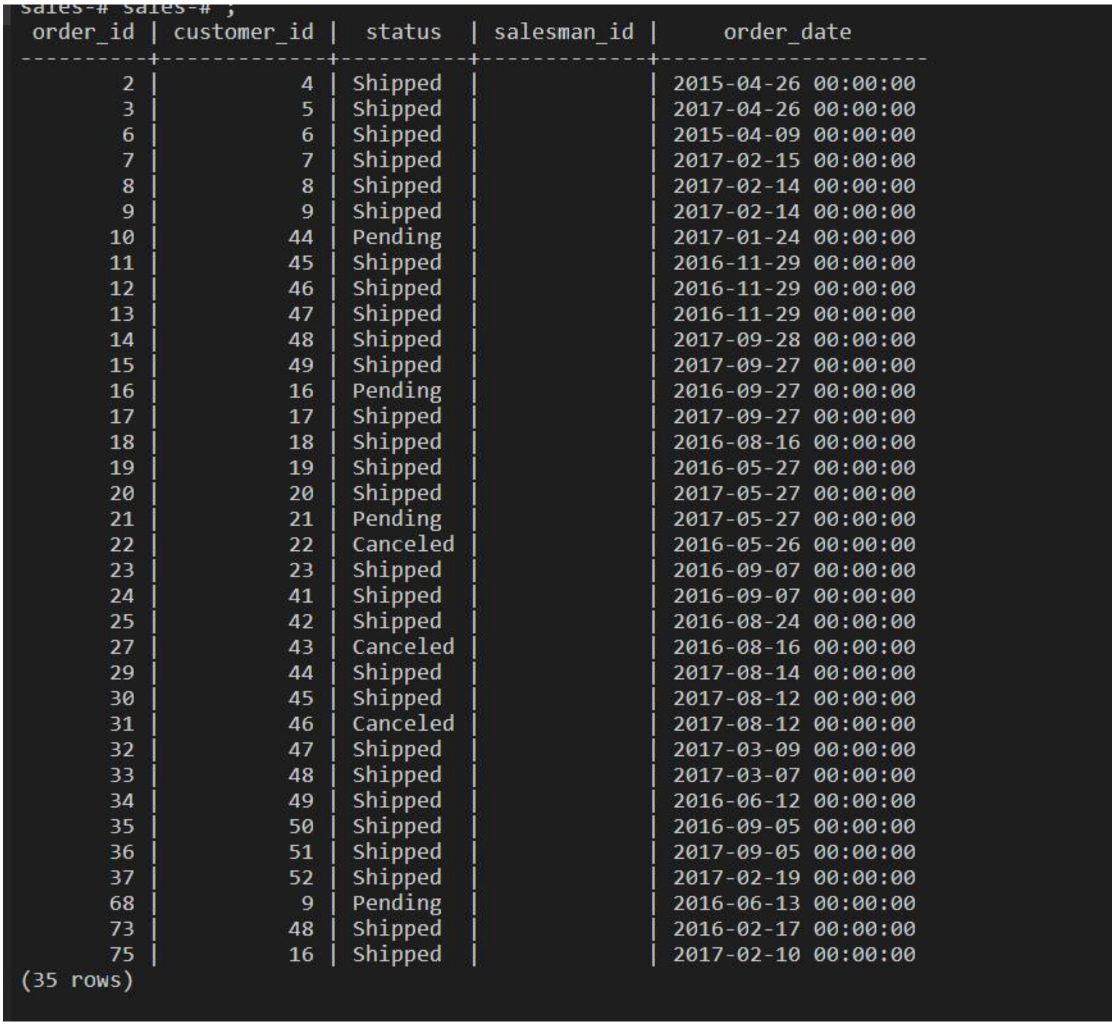
```sql
SELECT first_name
FROM contacts
WHERE first_name LIKE 'Je%' AND length(first_name)>=3 AND
length(last_name)>=3
AND length(email)>=3 AND length(phone)>=3;
```



```
sales=# SELECT first_name
FROM contacts
WHERE first_name LIKE 'Je%' AND length(first_name)>=3 AND length(last_name)>=3
AND length(email)>=3 AND length(phone)>=3sales-# sales-# sales-#
sales-# ;
 first_name
------------
 Jeni
 Jessika
 Jeri
 Jermaine
 Jeannie
 Jess
 Jerica
 Jen
(8 rows)
```

17. 查询订单表中所有没有销售员负责的订单

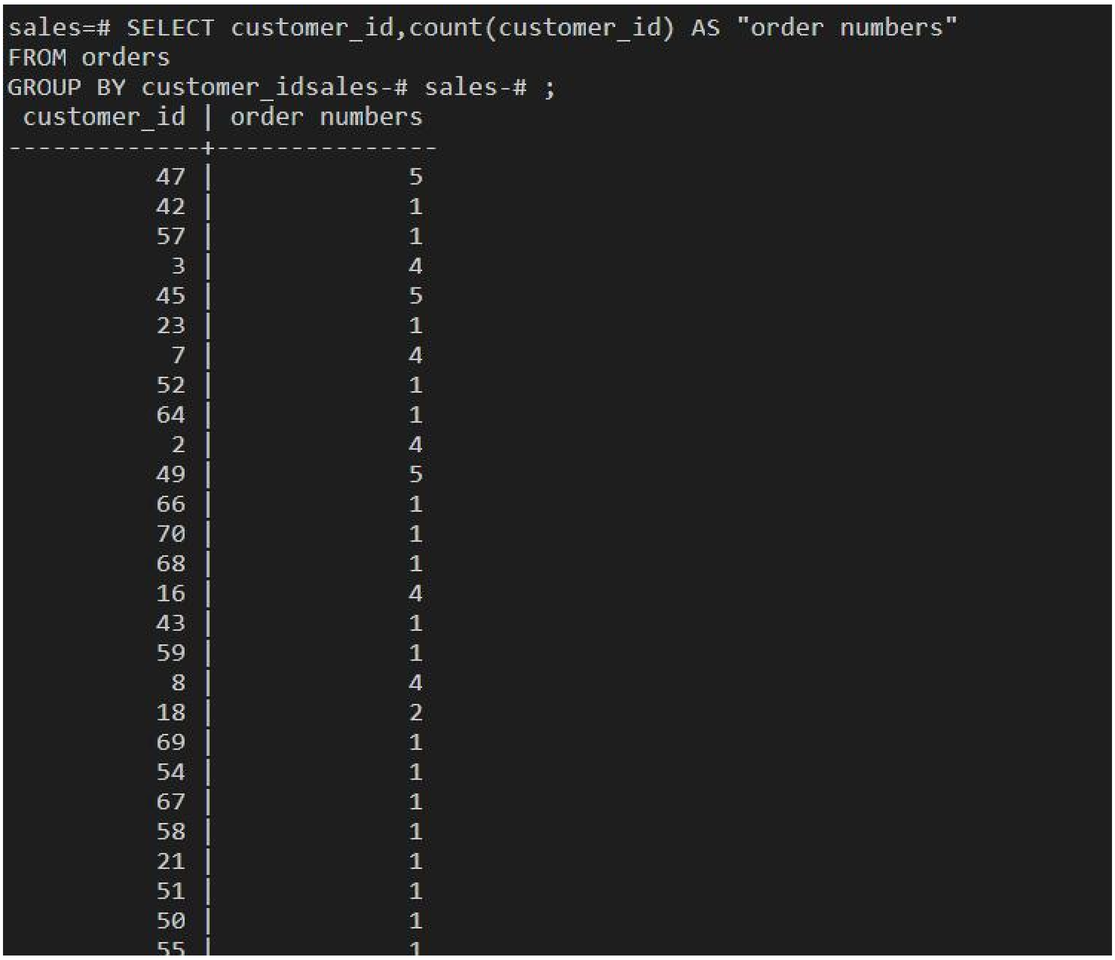
```sql
SELECT * FROM orders
WHERE salesman_id IS NULL;
```



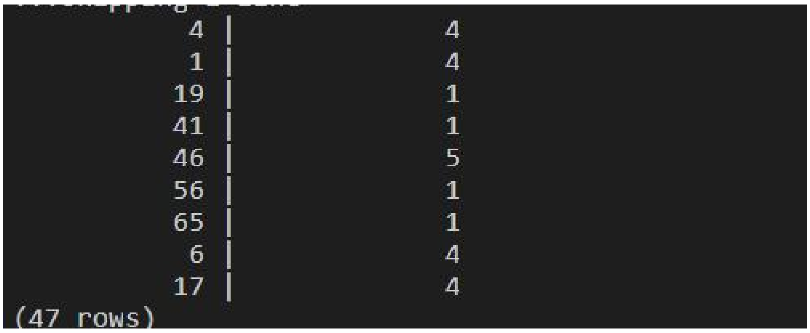
```
sales-# sales-# ;
 order_id | customer_id |  status   | salesman_id |      order_date
----------+-------------+-----------+-------------+---------------------
        2 |           4 | Shipped   |             | 2015-04-26 00:00:00
        3 |           5 | Shipped   |             | 2017-04-26 00:00:00
        6 |           6 | Shipped   |             | 2015-04-09 00:00:00
        7 |           7 | Shipped   |             | 2017-02-15 00:00:00
        8 |           8 | Shipped   |             | 2017-02-14 00:00:00
        9 |           9 | Shipped   |             | 2017-02-14 00:00:00
       10 |          44 | Pending   |             | 2017-01-24 00:00:00
       11 |          45 | Shipped   |             | 2016-11-29 00:00:00
       12 |          46 | Shipped   |             | 2016-11-29 00:00:00
       13 |          47 | Shipped   |             | 2016-11-29 00:00:00
       14 |          48 | Shipped   |             | 2017-09-28 00:00:00
       15 |          49 | Shipped   |             | 2017-09-27 00:00:00
       16 |          16 | Pending   |             | 2016-09-27 00:00:00
       17 |          17 | Shipped   |             | 2017-09-27 00:00:00
       18 |          18 | Shipped   |             | 2016-08-16 00:00:00
       19 |          19 | Shipped   |             | 2016-05-27 00:00:00
       20 |          20 | Shipped   |             | 2017-05-27 00:00:00
       21 |          21 | Pending   |             | 2017-05-27 00:00:00
       22 |          22 | Canceled  |             | 2016-05-26 00:00:00
       23 |          23 | Shipped   |             | 2016-09-07 00:00:00
       24 |          41 | Shipped   |             | 2016-09-07 00:00:00
       25 |          42 | Shipped   |             | 2016-08-24 00:00:00
       27 |          43 | Canceled  |             | 2016-08-16 00:00:00
       29 |          44 | Shipped   |             | 2017-08-14 00:00:00
       30 |          45 | Shipped   |             | 2017-08-12 00:00:00
       31 |          46 | Canceled  |             | 2017-08-12 00:00:00
       32 |          47 | Shipped   |             | 2017-03-09 00:00:00
       33 |          48 | Shipped   |             | 2017-03-07 00:00:00
       34 |          49 | Shipped   |             | 2016-06-12 00:00:00
       35 |          50 | Shipped   |             | 2016-09-05 00:00:00
       36 |          51 | Shipped   |             | 2017-09-05 00:00:00
       37 |          52 | Shipped   |             | 2017-02-19 00:00:00
       68 |           9 | Pending   |             | 2016-06-13 00:00:00
       73 |          48 | Shipped   |             | 2016-02-17 00:00:00
       75 |          16 | Shipped   |             | 2017-02-10 00:00:00
(35 rows)
```

18. 统计每个顾客的订单总数（查询订单表）

```
SELECT customer_id,count(customer_id) AS "order numbers"
FROM orders
GROUP BY customer_id;
```



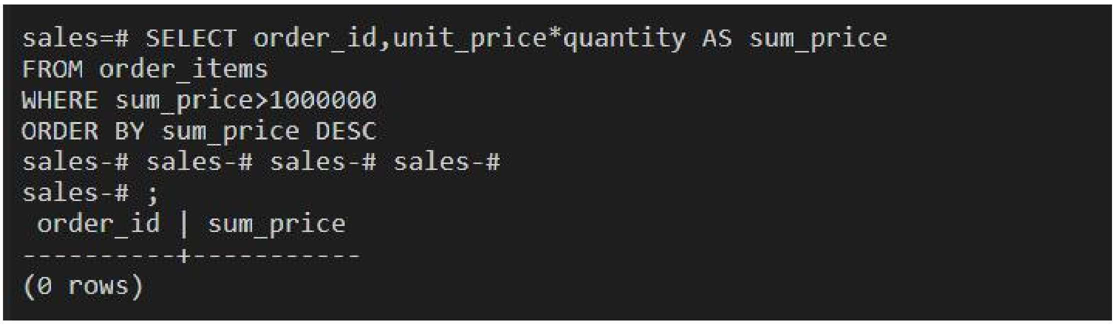
```
sales=# SELECT customer_id,count(customer_id) AS "order numbers"
FROM orders
GROUP BY customer_idsales-# sales-# ;
 customer_id | order numbers
-------------+---------------
          47 |             5
          42 |             1
          57 |             1
           3 |             4
          45 |             5
          23 |             1
           7 |             4
          52 |             1
          64 |             1
           2 |             4
          49 |             5
          66 |             1
          70 |             1
          68 |             1
          16 |             4
          43 |             1
          59 |             1
           8 |             4
          18 |             2
          69 |             1
          54 |             1
          67 |             1
          58 |             1
          21 |             1
          51 |             1
          50 |             1
          55 |             1
```



```
           4 |             4
           1 |             4
          19 |             1
          41 |             1
          46 |             5
          56 |             1
          65 |             1
           6 |             4
          17 |             4
(47 rows)
```

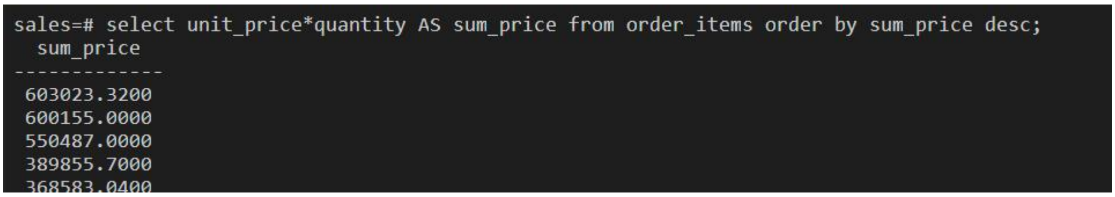
19. 统计每个订单的总价格大于 1000000 的订单号和总价格，并按总价格降序显示结果

```
SELECT order_id,unit_price*quantity AS sum_price
FROM order_items
WHERE sum_price>1000000
ORDER BY sum_price DESC;
```

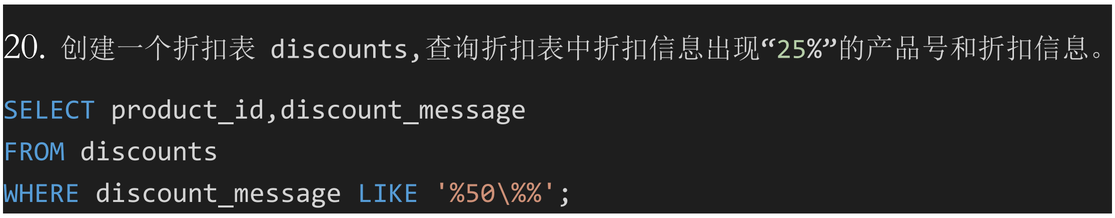


```
sales=# SELECT order_id,unit_price*quantity AS sum_price
FROM order_items
WHERE sum_price>1000000
ORDER BY sum_price DESC
sales-# sales-# sales-# sales-#
sales-# ;
 order_id | sum_price
----------+-----------
(0 rows)
```

无总值超过 1000000 的订单

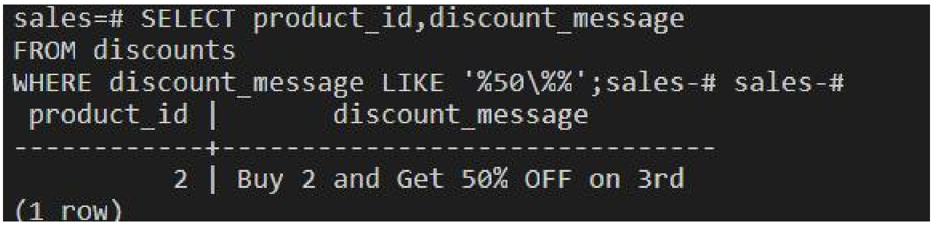


```
sales=# select unit_price*quantity AS sum_price from order_items order by sum_price desc;
  sum_price
-------------
 603023.3200
 600155.0000
 550487.0000
 389855.7000
 368583.0400
```



20. 创建一个折扣表 discounts,查询折扣表中折扣信息出现“25%”的产品号和折扣信息。

```
SELECT product_id,discount_message
FROM discounts
WHERE discount_message LIKE '%50\%%';
```



```
sales=# SELECT product_id,discount_message
FROM discounts
WHERE discount_message LIKE '%50\%%';sales-# sales-#
 product_id |       discount_message
------------+-------------------------------
          2 | Buy 2 and Get 50% OFF on 3rd
(1 row)
```

# 3.实验总结

## 3.1 完成的工作

1.熟悉实验二中的操作，如数据库的创建，表的创建，约束，数据的插入等；

2.学习 sql 语句的基本语法和用法，对数据进行查询。

## 3.2 对实验的认识

熟练掌握 openGauss 单表查询的语法结构及其使用方法；

掌握设计正确查询语句以实现查询要求的方法如普通单表查询、模糊查询、聚集函数、分组统计和排序;

掌握 Group by 的使用 ;

掌握 Order by 的使用 ;

掌握使用 DISTINCT 实现查询结果的去重方法;

掌握空值 NULL 的使用方法 ;

掌握查询过程中别名的使用方法 ;

掌握自身连接的使用方法;

## 3.3 遇到的困难及解决方法

无