数据库实验报告

李甘 2023202296

实验一 基于文件系统的商城库存管理系统

1. 实现方法

系统以CSV文件作为持久化介质，商品与进销记录分别存储于products.csv与records.csv；内存中加载为结构体列表，按需检索、排序与统计，商品删除采用逻辑删除。所有增删改查通过顺序读写与追加实现，并对时间与数量等输入做基本校验。

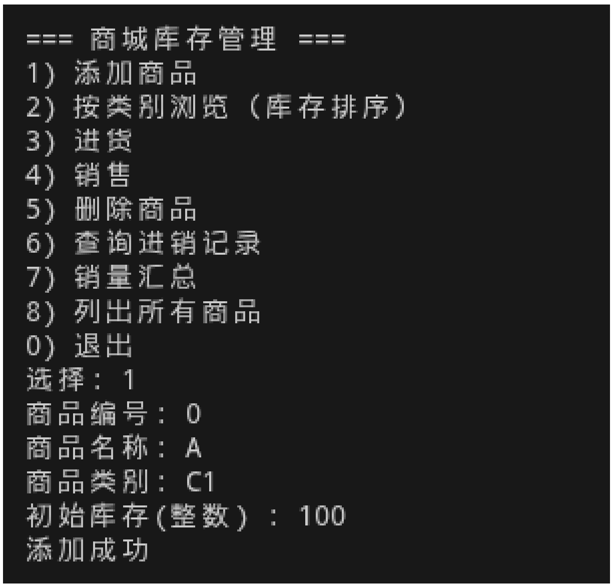
商品数据结构

字段	含义	示例
id	商品编号（唯一）	P001
name	商品名称	Apple
category	商品类别	Fruit
stock	当前库存	120
deleted	逻辑删除标记	false

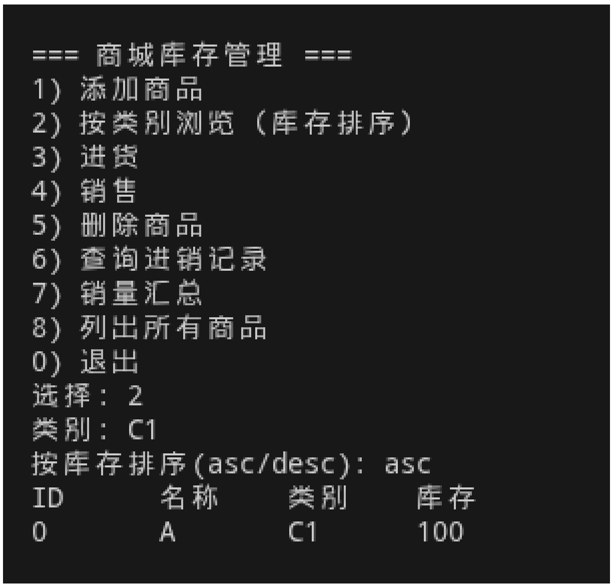
进销记录数据结构

字段	含义	示例
productId	商品编号	P001
productName	商品名称（冗余便于历史可读）	Apple
type	记录类型（进货/销售）	Purchase
operatorName	操作人	Alice
timestamp	时间（YYYY-MM-DD HH:MM:SS）	2025-10-10 09:30:00
quantity	数量（正整数）	50

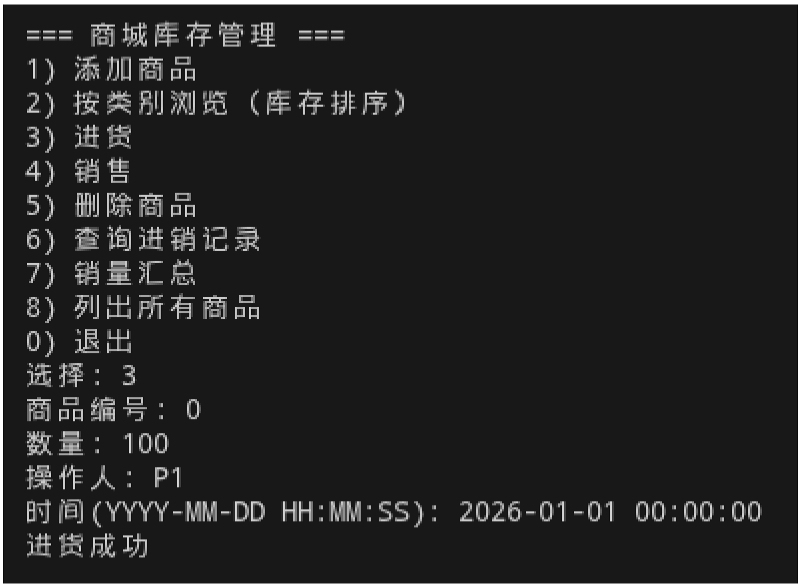
2. 运行截图



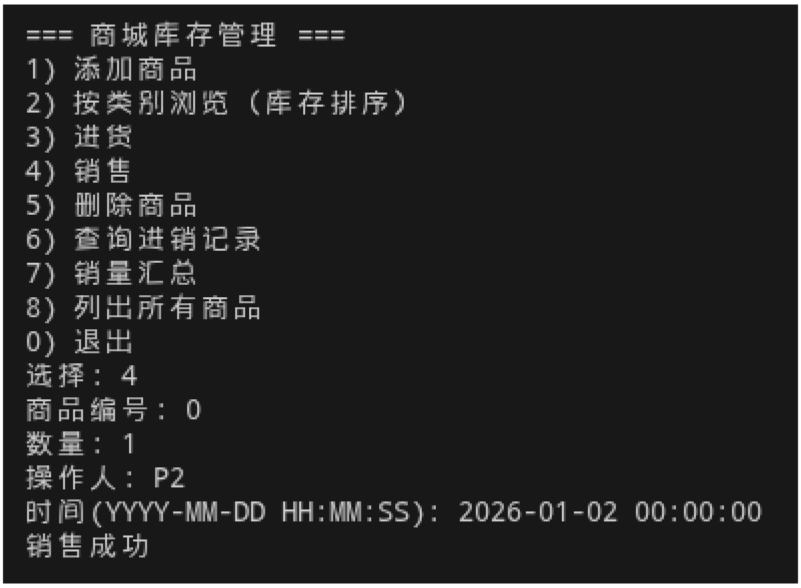
```
=== 商城库存管理 ===
1) 添加商品
2) 按类别浏览（库存排序）
3) 进货
4) 销售
5) 删除商品
6) 查询进销记录
7) 销量汇总
8) 列出所有商品
0) 退出
选择：1
商品编号：0
商品名称：A
商品类别：C1
初始库存(整数)：100
添加成功
```



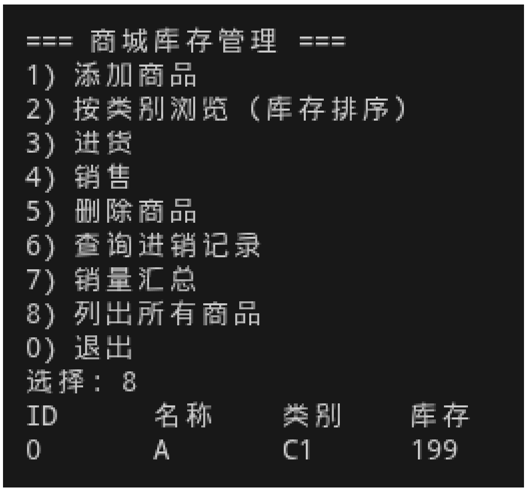
```
=== 商城库存管理 ===
1) 添加商品
2) 按类别浏览（库存排序）
3) 进货
4) 销售
5) 删除商品
6) 查询进销记录
7) 销量汇总
8) 列出所有商品
0) 退出
选择：2
类别：C1
按库存排序(asc/desc): asc
ID        名称      类别      库存
0         A         C1        100
```



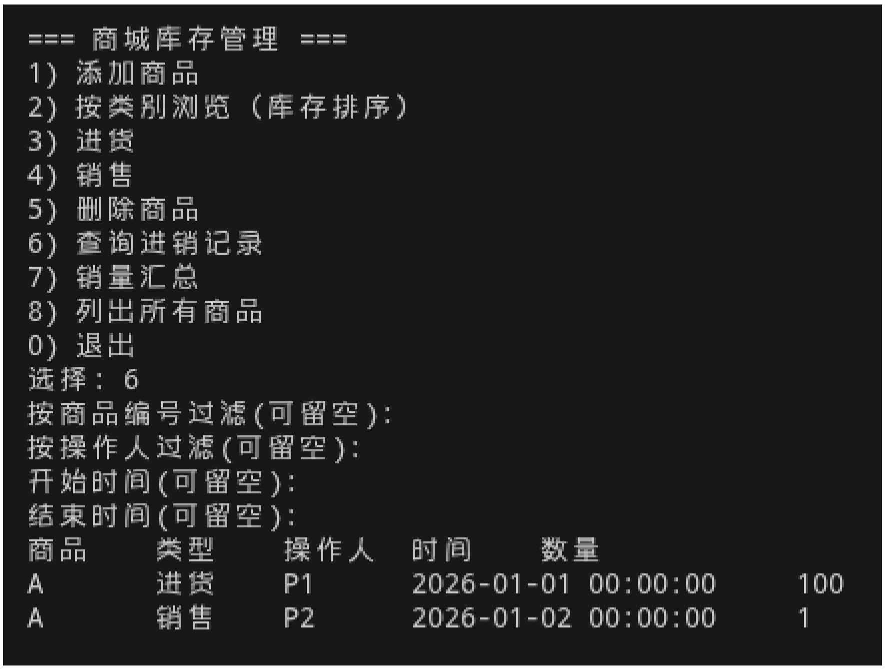
```
=== 商城库存管理 ===
1) 添加商品
2) 按类别浏览（库存排序）
3) 进货
4) 销售
5) 删除商品
6) 查询进销记录
7) 销量汇总
8) 列出所有商品
0) 退出
选择：3
商品编号：0
数量：100
操作人：P1
时间(YYYY-MM-DD HH:MM:SS): 2026-01-01 00:00:00
进货成功
```



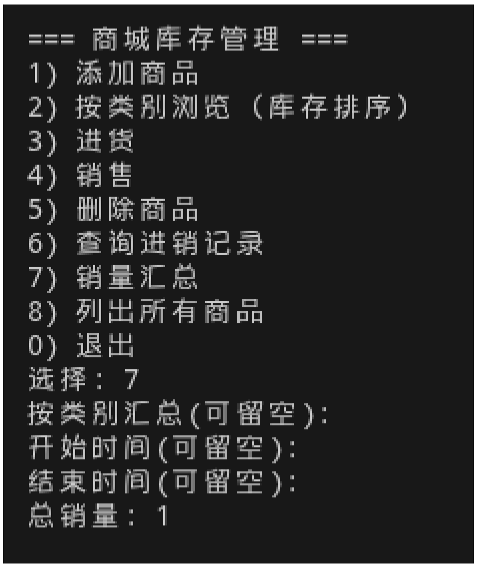
```
=== 商城库存管理 ===
1） 添加商品
2） 按类别浏览 （库存排序）
3） 进货
4） 销售
5） 删除商品
6） 查询进销记录
7） 销量汇总
8） 列出所有商品
0） 退出
选择： 4
商品编号： 0
数量： 1
操作人： P2
时间(YYYY-MM-DD HH:MM:SS)： 2026-01-02 00:00:00
销售成功
```



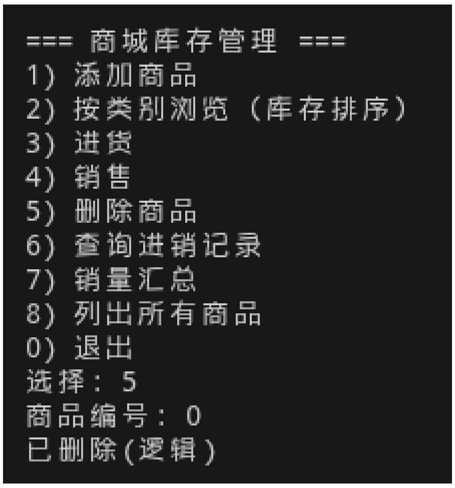
```
=== 商城库存管理 ===
1） 添加商品
2） 按类别浏览 （库存排序）
3） 进货
4） 销售
5） 删除商品
6） 查询进销记录
7） 销量汇总
8） 列出所有商品
0） 退出
选择： 8
ID      名称    类别    库存
0       A       C1      199
```



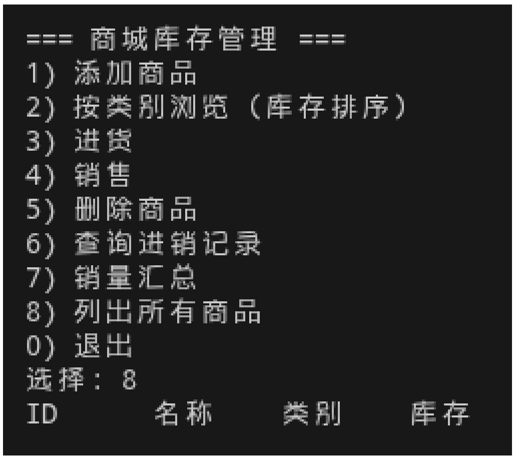
```
=== 商城库存管理 ===
1） 添加商品
2） 按类别浏览 （库存排序）
3） 进货
4） 销售
5） 删除商品
6） 查询进销记录
7） 销量汇总
8） 列出所有商品
0） 退出
选择： 6
按商品编号过滤(可留空):
按操作人过滤(可留空):
开始时间(可留空):
结束时间(可留空):
商品    类型    操作人  时间    数量
A       进货    P1      2026-01-01 00:00:00     100
A       销售    P2      2026-01-02 00:00:00     1
```



```
=== 商城库存管理 ===
1) 添加商品
2) 按类别浏览 (库存排序)
3) 进货
4) 销售
5) 删除商品
6) 查询进销记录
7) 销量汇总
8) 列出所有商品
0) 退出
选择: 7
按类别汇总(可留空):
开始时间(可留空):
结束时间(可留空):
总销量: 1
```



```
=== 商城库存管理 ===
1) 添加商品
2) 按类别浏览 (库存排序)
3) 进货
4) 销售
5) 删除商品
6) 查询进销记录
7) 销量汇总
8) 列出所有商品
0) 退出
选择: 5
商品编号: 0
已删除(逻辑)
```



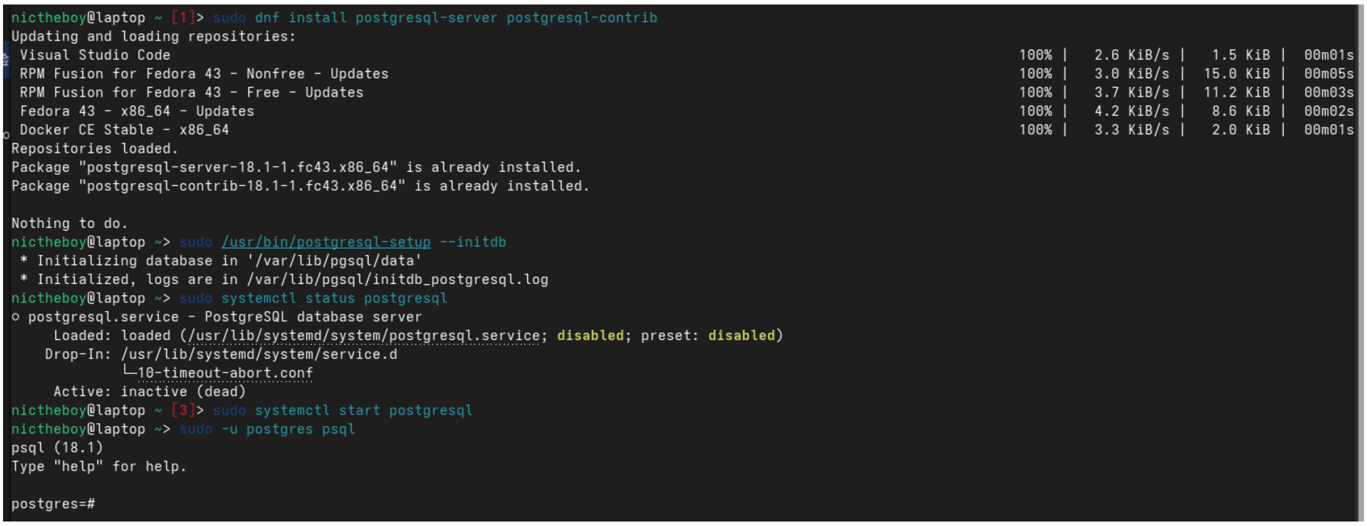
```
=== 商城库存管理 ===
1) 添加商品
2) 按类别浏览 (库存排序)
3) 进货
4) 销售
5) 删除商品
6) 查询进销记录
7) 销量汇总
8) 列出所有商品
0) 退出
选择: 8
ID      名称      类别      库存
```

实验二 数据库系统的使用

我在我的安装有 Fedora Linux 的笔记本电脑上用 PostgreSQL 完成了下面的实验。

1. 安装并启动数据库

```
sudo dnf install postgresql-server postgresql-contrib
sudo /usr/bin/postgresql-setup --initdb
sudo systemctl status postgresql
sudo systemctl start postgresql
sudo -u postgres psql
```

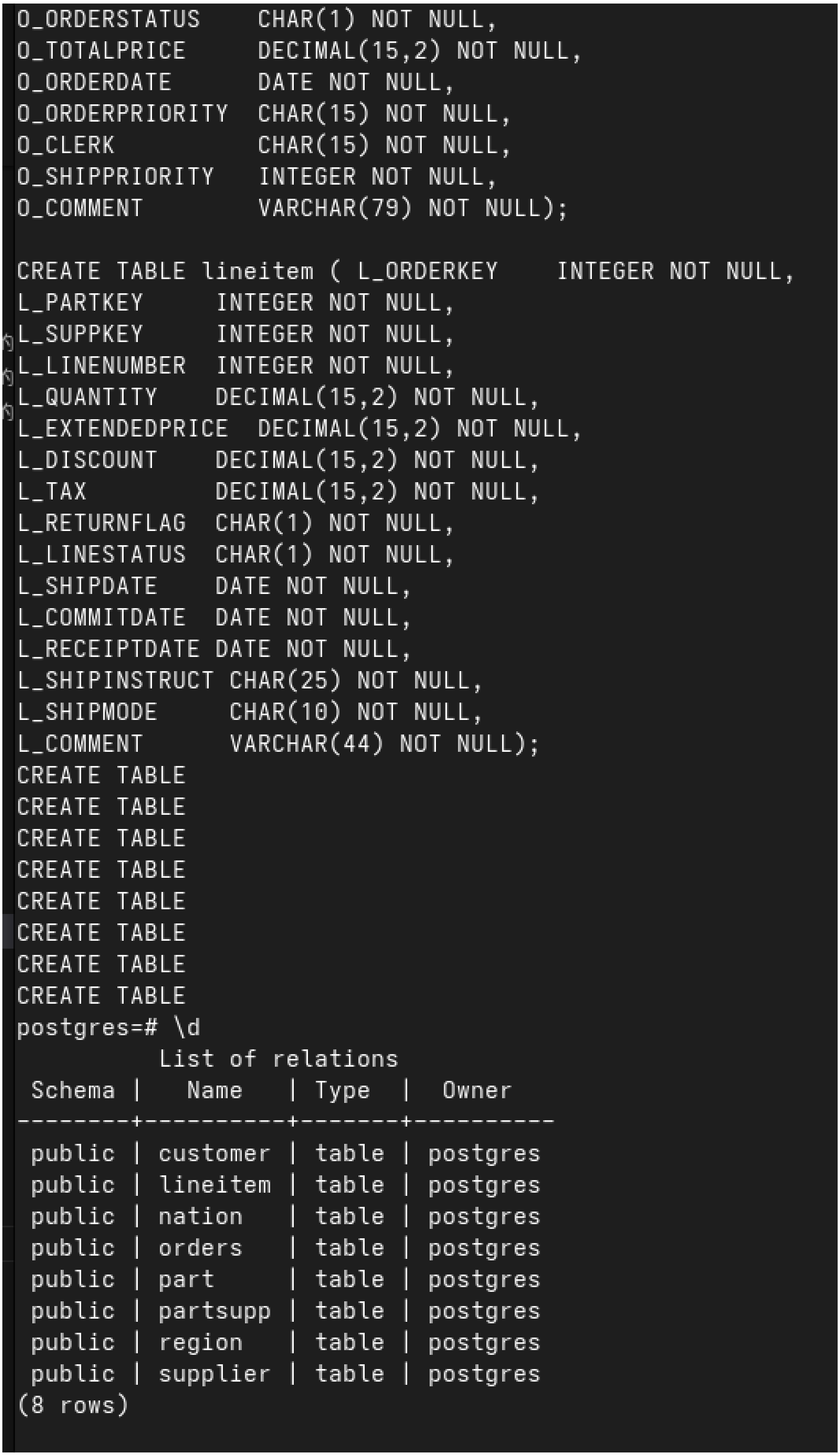


```
nictheboy@laptop ~ [1]> sudo dnf install postgresql-server postgresql-contrib
Updating and loading repositories:
 Visual Studio Code                                                        100% |   2.6 KiB/s |   1.5 KiB |  00m01s
 RPM Fusion for Fedora 43 - Nonfree - Updates                              100% |   3.0 KiB/s |  15.0 KiB |  00m05s
 RPM Fusion for Fedora 43 - Free - Updates                                 100% |   3.7 KiB/s |  11.2 KiB |  00m03s
 Fedora 43 - x86_64 - Updates                                              100% |   4.2 KiB/s |   8.6 KiB |  00m02s
 Docker CE Stable - x86_64                                                 100% |   3.3 KiB/s |   2.0 KiB |  00m01s
Repositories loaded.
Package "postgresql-server-18.1-1.fc43.x86_64" is already installed.
Package "postgresql-contrib-18.1-1.fc43.x86_64" is already installed.

Nothing to do.
nictheboy@laptop ~> sudo /usr/bin/postgresql-setup --initdb
 * Initializing database in '/var/lib/pgsql/data'
 * Initialized, logs are in /var/lib/pgsql/initdb_postgresql.log
nictheboy@laptop ~> sudo systemctl status postgresql
○ postgresql.service - PostgreSQL database server
     Loaded: loaded (/usr/lib/systemd/system/postgresql.service; disabled; preset: disabled)
    Drop-In: /usr/lib/systemd/system/service.d
             └─10-timeout-abort.conf
     Active: inactive (dead)
nictheboy@laptop ~ [3]> sudo systemctl start postgresql
nictheboy@laptop ~> sudo -u postgres psql
psql (18.1)
Type "help" for help.

postgres=#
```

2. 创建表结构

```
CREATE TABLE nation  ( N_NATIONKEY  INTEGER NOT NULL,
N_NAME       CHAR(25) NOT NULL,
N_REGIONKEY  INTEGER NOT NULL,
N_COMMENT    VARCHAR(152));

CREATE TABLE region  ( R_REGIONKEY  INTEGER NOT NULL,
R_NAME       CHAR(25) NOT NULL,
R_COMMENT    VARCHAR(152));

CREATE TABLE part  ( P_PARTKEY     INTEGER NOT NULL,
P_NAME        VARCHAR(55) NOT NULL,
P_MFGR        CHAR(25) NOT NULL,
P_BRAND       CHAR(10) NOT NULL,
P_TYPE        VARCHAR(25) NOT NULL,
P_SIZE        INTEGER NOT NULL,
P_CONTAINER   CHAR(10) NOT NULL,
P_RETAILPRICE DECIMAL(15,2) NOT NULL,
P_COMMENT     VARCHAR(23) NOT NULL );

CREATE TABLE supplier ( S_SUPPKEY     INTEGER NOT NULL,
S_NAME        CHAR(25) NOT NULL,
S_ADDRESS     VARCHAR(40) NOT NULL,
S_NATIONKEY   INTEGER NOT NULL,
S_PHONE       CHAR(15) NOT NULL,
S_ACCTBAL     DECIMAL(15,2) NOT NULL,
S_COMMENT     VARCHAR(101) NOT NULL);

CREATE TABLE partsupp ( PS_PARTKEY     INTEGER NOT NULL,
PS_SUPPKEY     INTEGER NOT NULL,
PS_AVAILQTY    INTEGER NOT NULL,
PS_SUPPLYCOST  DECIMAL(15,2)  NOT NULL,
PS_COMMENT     VARCHAR(199) NOT NULL );
```

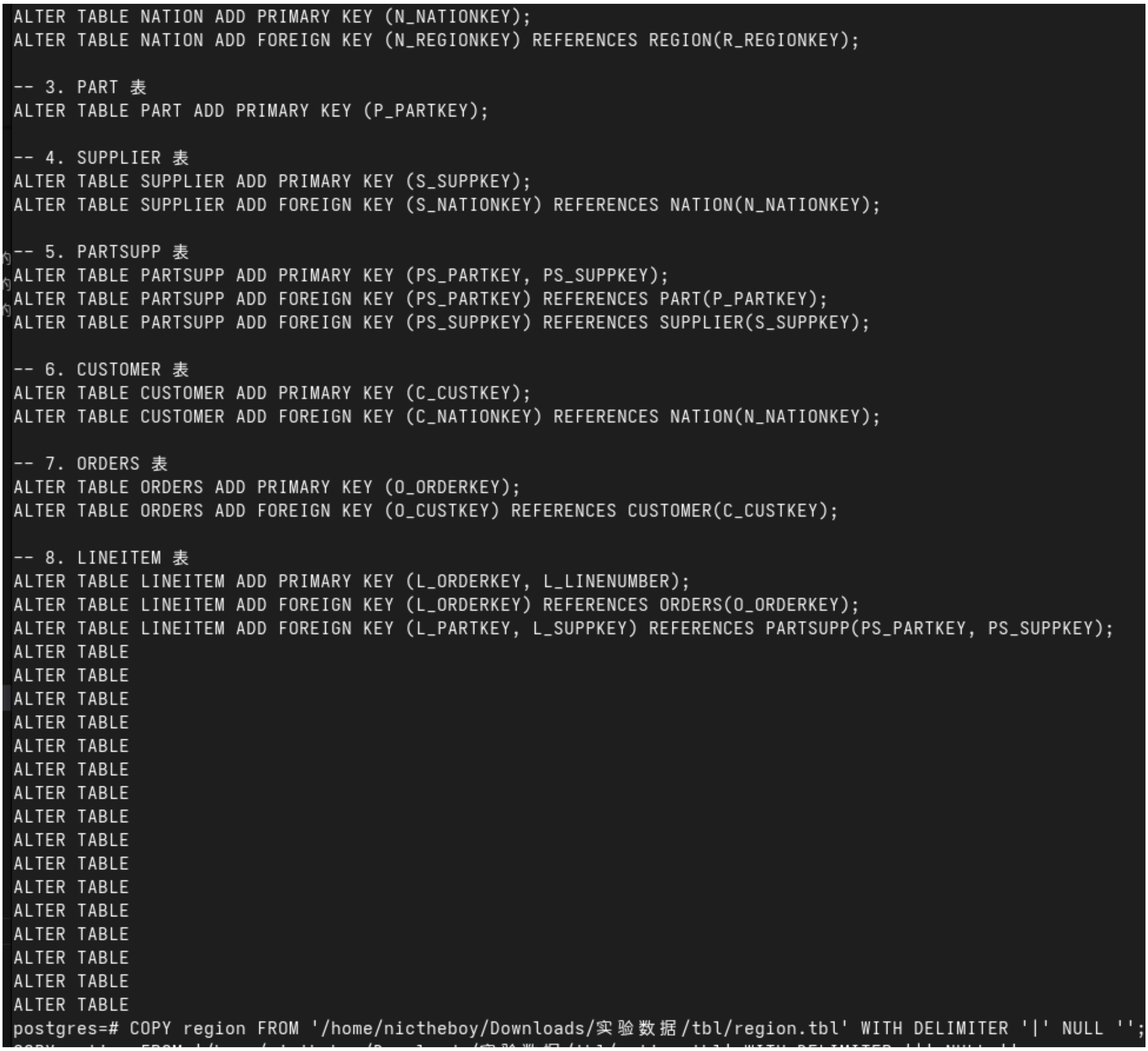
```
CREATE TABLE customer ( C_CUSTKEY     INTEGER NOT NULL,
C_NAME        VARCHAR(25) NOT NULL,
C_ADDRESS     VARCHAR(40) NOT NULL,
C_NATIONKEY   INTEGER NOT NULL,
C_PHONE       CHAR(15) NOT NULL,
C_ACCTBAL     DECIMAL(15,2)   NOT NULL,
C_MKTSEGMENT  CHAR(10) NOT NULL,
C_COMMENT     VARCHAR(117) NOT NULL);

CREATE TABLE orders  ( O_ORDERKEY       INTEGER NOT NULL,
O_CUSTKEY        INTEGER NOT NULL,
O_ORDERSTATUS    CHAR(1) NOT NULL,
O_TOTALPRICE     DECIMAL(15,2) NOT NULL,
O_ORDERDATE      DATE NOT NULL,
O_ORDERPRIORITY  CHAR(15) NOT NULL,
O_CLERK          CHAR(15) NOT NULL,
O_SHIPPRIORITY   INTEGER NOT NULL,
O_COMMENT        VARCHAR(79) NOT NULL);

CREATE TABLE lineitem ( L_ORDERKEY    INTEGER NOT NULL,
L_PARTKEY     INTEGER NOT NULL,
L_SUPPKEY     INTEGER NOT NULL,
L_LINENUMBER  INTEGER NOT NULL,
L_QUANTITY    DECIMAL(15,2) NOT NULL,
L_EXTENDEDPRICE  DECIMAL(15,2) NOT NULL,
L_DISCOUNT    DECIMAL(15,2) NOT NULL,
L_TAX         DECIMAL(15,2) NOT NULL,
L_RETURNFLAG  CHAR(1) NOT NULL,
L_LINESTATUS  CHAR(1) NOT NULL,
L_SHIPDATE    DATE NOT NULL,
L_COMMITDATE  DATE NOT NULL,
L_RECEIPTDATE DATE NOT NULL,
L_SHIPINSTRUCT CHAR(25) NOT NULL,
L_SHIPMODE     CHAR(10) NOT NULL,
L_COMMENT      VARCHAR(44) NOT NULL);
```



```
O_ORDERSTATUS    CHAR(1) NOT NULL,
O_TOTALPRICE     DECIMAL(15,2) NOT NULL,
O_ORDERDATE      DATE NOT NULL,
O_ORDERPRIORITY  CHAR(15) NOT NULL,
O_CLERK          CHAR(15) NOT NULL,
O_SHIPPRIORITY   INTEGER NOT NULL,
O_COMMENT        VARCHAR(79) NOT NULL);

CREATE TABLE lineitem ( L_ORDERKEY    INTEGER NOT NULL,
L_PARTKEY     INTEGER NOT NULL,
L_SUPPKEY     INTEGER NOT NULL,
L_LINENUMBER  INTEGER NOT NULL,
L_QUANTITY    DECIMAL(15,2) NOT NULL,
L_EXTENDEDPRICE  DECIMAL(15,2) NOT NULL,
L_DISCOUNT    DECIMAL(15,2) NOT NULL,
L_TAX         DECIMAL(15,2) NOT NULL,
L_RETURNFLAG  CHAR(1) NOT NULL,
L_LINESTATUS  CHAR(1) NOT NULL,
L_SHIPDATE    DATE NOT NULL,
L_COMMITDATE  DATE NOT NULL,
L_RECEIPTDATE DATE NOT NULL,
L_SHIPINSTRUCT CHAR(25) NOT NULL,
L_SHIPMODE     CHAR(10) NOT NULL,
L_COMMENT      VARCHAR(44) NOT NULL);
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
postgres=# \d
          List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | customer | table | postgres
 public | lineitem | table | postgres
 public | nation   | table | postgres
 public | orders   | table | postgres
 public | part     | table | postgres
 public | partsupp | table | postgres
 public | region   | table | postgres
 public | supplier | table | postgres
(8 rows)
```

3. 添加主键和外键

```sql
-- 1. REGION 表
ALTER TABLE REGION ADD PRIMARY KEY (R_REGIONKEY);

-- 2. NATION 表
ALTER TABLE NATION ADD PRIMARY KEY (N_NATIONKEY);
ALTER TABLE NATION ADD FOREIGN KEY (N_REGIONKEY) REFERENCES
REGION(R_REGIONKEY);

-- 3. PART 表
ALTER TABLE PART ADD PRIMARY KEY (P_PARTKEY);

-- 4. SUPPLIER 表
ALTER TABLE SUPPLIER ADD PRIMARY KEY (S_SUPPKEY);
ALTER TABLE SUPPLIER ADD FOREIGN KEY (S_NATIONKEY) REFERENCES
NATION(N_NATIONKEY);

-- 5. PARTSUPP 表
ALTER TABLE PARTSUPP ADD PRIMARY KEY (PS_PARTKEY, PS_SUPPKEY);
ALTER TABLE PARTSUPP ADD FOREIGN KEY (PS_PARTKEY) REFERENCES
PART(P_PARTKEY);
ALTER TABLE PARTSUPP ADD FOREIGN KEY (PS_SUPPKEY) REFERENCES
SUPPLIER(S_SUPPKEY);

-- 6. CUSTOMER 表
ALTER TABLE CUSTOMER ADD PRIMARY KEY (C_CUSTKEY);
ALTER TABLE CUSTOMER ADD FOREIGN KEY (C_NATIONKEY) REFERENCES
NATION(N_NATIONKEY);

-- 7. ORDERS 表
ALTER TABLE ORDERS ADD PRIMARY KEY (O_ORDERKEY);
ALTER TABLE ORDERS ADD FOREIGN KEY (O_CUSTKEY) REFERENCES
CUSTOMER(C_CUSTKEY);

-- 8. LINEITEM 表
ALTER TABLE LINEITEM ADD PRIMARY KEY (L_ORDERKEY, L_LINENUMBER);
ALTER TABLE LINEITEM ADD FOREIGN KEY (L_ORDERKEY) REFERENCES
ORDERS(O_ORDERKEY);
ALTER TABLE LINEITEM ADD FOREIGN KEY (L_PARTKEY, L_SUPPKEY) REFERENCES
PARTSUPP(PS_PARTKEY, PS_SUPPKEY);
```



```
ALTER TABLE NATION ADD PRIMARY KEY (N_NATIONKEY);
ALTER TABLE NATION ADD FOREIGN KEY (N_REGIONKEY) REFERENCES REGION(R_REGIONKEY);

-- 3. PART 表
ALTER TABLE PART ADD PRIMARY KEY (P_PARTKEY);

-- 4. SUPPLIER 表
ALTER TABLE SUPPLIER ADD PRIMARY KEY (S_SUPPKEY);
ALTER TABLE SUPPLIER ADD FOREIGN KEY (S_NATIONKEY) REFERENCES NATION(N_NATIONKEY);

-- 5. PARTSUPP 表
ALTER TABLE PARTSUPP ADD PRIMARY KEY (PS_PARTKEY, PS_SUPPKEY);
ALTER TABLE PARTSUPP ADD FOREIGN KEY (PS_PARTKEY) REFERENCES PART(P_PARTKEY);
ALTER TABLE PARTSUPP ADD FOREIGN KEY (PS_SUPPKEY) REFERENCES SUPPLIER(S_SUPPKEY);

-- 6. CUSTOMER 表
ALTER TABLE CUSTOMER ADD PRIMARY KEY (C_CUSTKEY);
ALTER TABLE CUSTOMER ADD FOREIGN KEY (C_NATIONKEY) REFERENCES NATION(N_NATIONKEY);

-- 7. ORDERS 表
ALTER TABLE ORDERS ADD PRIMARY KEY (O_ORDERKEY);
ALTER TABLE ORDERS ADD FOREIGN KEY (O_CUSTKEY) REFERENCES CUSTOMER(C_CUSTKEY);

-- 8. LINEITEM 表
ALTER TABLE LINEITEM ADD PRIMARY KEY (L_ORDERKEY, L_LINENUMBER);
ALTER TABLE LINEITEM ADD FOREIGN KEY (L_ORDERKEY) REFERENCES ORDERS(O_ORDERKEY);
ALTER TABLE LINEITEM ADD FOREIGN KEY (L_PARTKEY, L_SUPPKEY) REFERENCES PARTSUPP(PS_PARTKEY, PS_SUPPKEY);
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
postgres=# COPY region FROM '/home/nictheboy/Downloads/实验数据/tbl/region.tbl' WITH DELIMITER '|' NULL '';
```

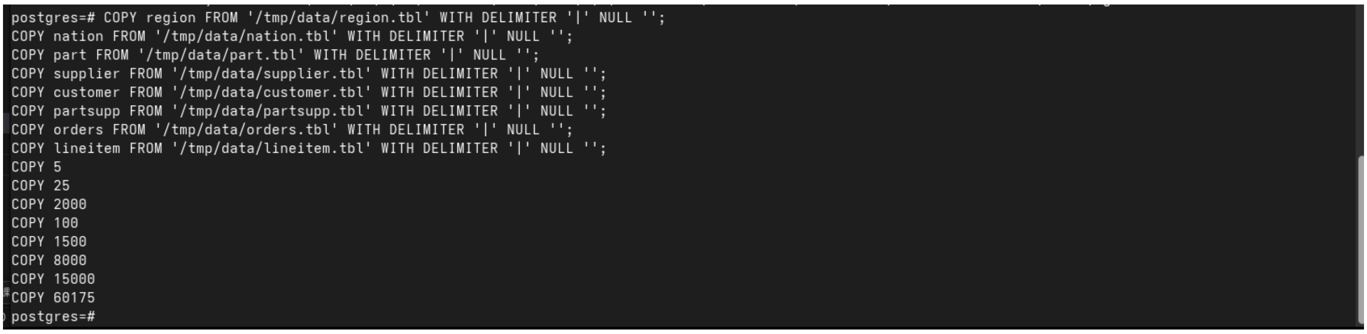
4. 导入数据

先把数据文件末尾的 | 去掉：

```
sed -i 's/|$//' *.tbl
```

然后导入数据：

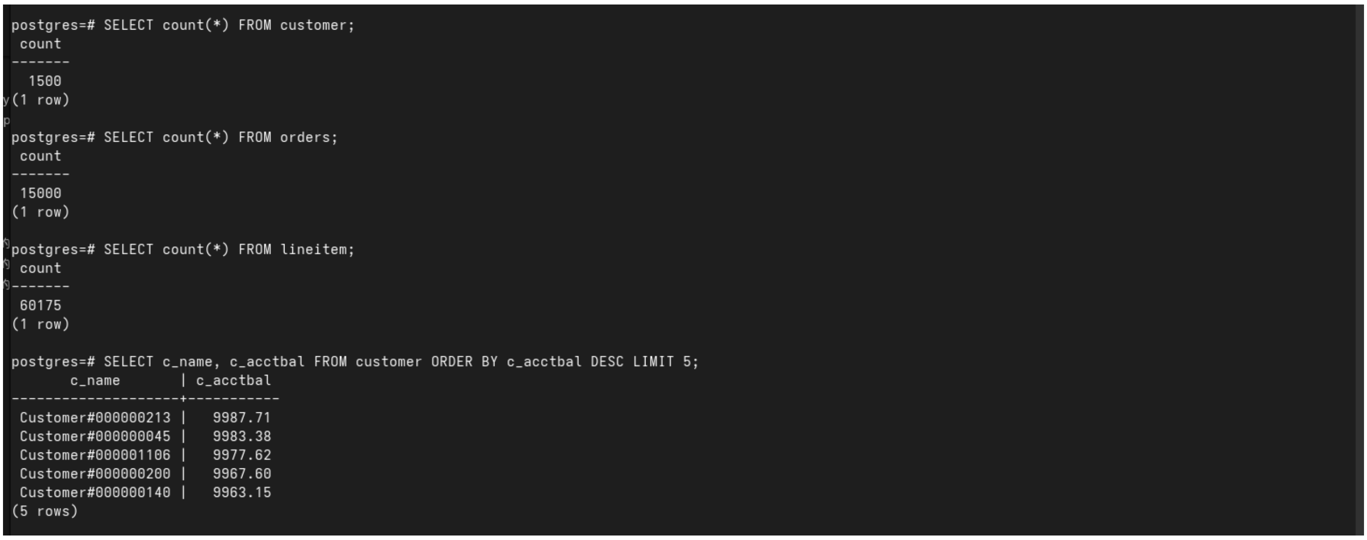
```
COPY region FROM '/tmp/data/region.tbl' WITH DELIMITER '|' NULL '';
COPY nation FROM '/tmp/data/nation.tbl' WITH DELIMITER '|' NULL '';
COPY part FROM '/tmp/data/part.tbl' WITH DELIMITER '|' NULL '';
COPY supplier FROM '/tmp/data/supplier.tbl' WITH DELIMITER '|' NULL '';
COPY customer FROM '/tmp/data/customer.tbl' WITH DELIMITER '|' NULL '';
COPY partsupp FROM '/tmp/data/partsupp.tbl' WITH DELIMITER '|' NULL '';
COPY orders FROM '/tmp/data/orders.tbl' WITH DELIMITER '|' NULL '';
COPY lineitem FROM '/tmp/data/lineitem.tbl' WITH DELIMITER '|' NULL '';
```



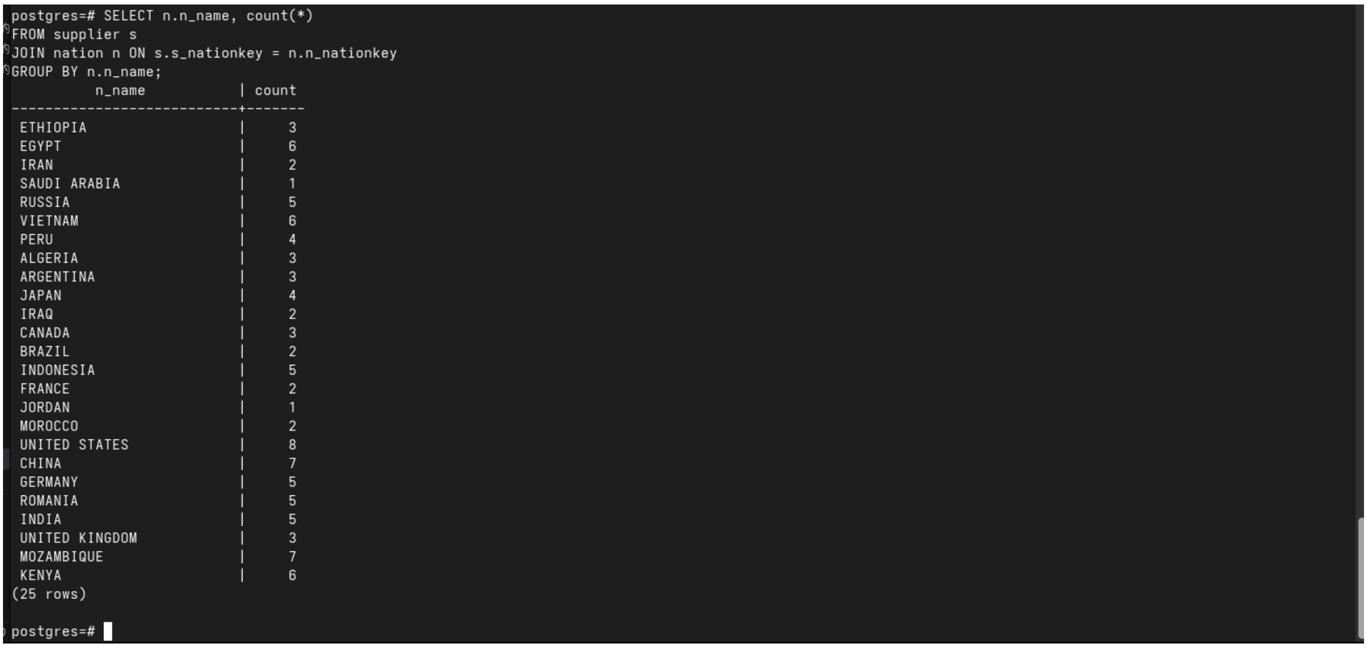
```
postgres=# COPY region FROM '/tmp/data/region.tbl' WITH DELIMITER '|' NULL '';
COPY nation FROM '/tmp/data/nation.tbl' WITH DELIMITER '|' NULL '';
COPY part FROM '/tmp/data/part.tbl' WITH DELIMITER '|' NULL '';
COPY supplier FROM '/tmp/data/supplier.tbl' WITH DELIMITER '|' NULL '';
COPY customer FROM '/tmp/data/customer.tbl' WITH DELIMITER '|' NULL '';
COPY partsupp FROM '/tmp/data/partsupp.tbl' WITH DELIMITER '|' NULL '';
COPY orders FROM '/tmp/data/orders.tbl' WITH DELIMITER '|' NULL '';
COPY lineitem FROM '/tmp/data/lineitem.tbl' WITH DELIMITER '|' NULL '';
COPY 5
COPY 25
COPY 2000
COPY 100
COPY 1500
COPY 8000
COPY 15000
COPY 60175
postgres=#
```

5. 验证和查询

```sql
SELECT count(*) FROM customer;
SELECT count(*) FROM orders;
SELECT count(*) FROM lineitem;

-- 查询余额最高的前5个消费者
SELECT c_name, c_acctbal FROM customer ORDER BY c_acctbal DESC LIMIT 5;

-- 带 Join 的查询：查询每个国家的供应商数量
SELECT n.n_name, count(*)
FROM supplier s
JOIN nation n ON s.s_nationkey = n.n_nationkey
GROUP BY n.n_name;
```



```
postgres=# SELECT count(*) FROM customer;
 count
-------
  1500
(1 row)

postgres=# SELECT count(*) FROM orders;
 count
-------
 15000
(1 row)

postgres=# SELECT count(*) FROM lineitem;
 count
-------
 60175
(1 row)

postgres=# SELECT c_name, c_acctbal FROM customer ORDER BY c_acctbal DESC LIMIT 5;
       c_name       | c_acctbal
--------------------+-----------
 Customer#000000213 |   9987.71
 Customer#000000045 |   9983.38
 Customer#000001106 |   9977.62
 Customer#000000200 |   9967.60
 Customer#000000140 |   9963.15
(5 rows)
```



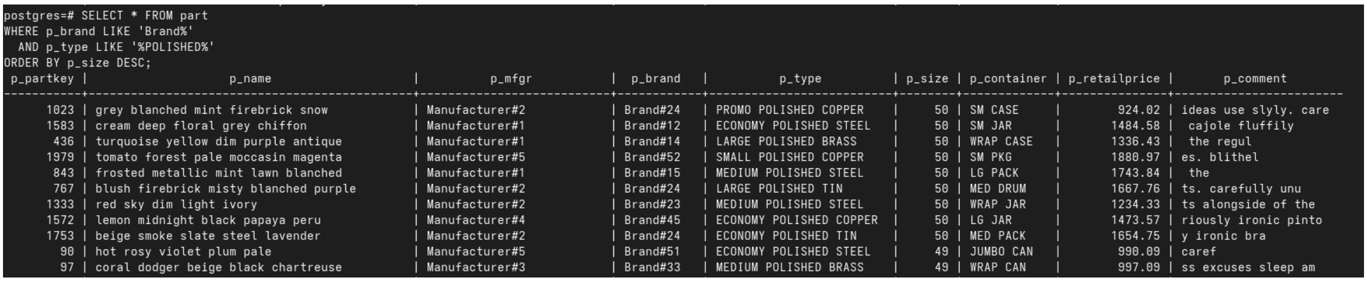
```
postgres=# SELECT n.n_name, count(*)
FROM supplier s
JOIN nation n ON s.s_nationkey = n.n_nationkey
GROUP BY n.n_name;
          n_name           | count
---------------------------+-------
 ETHIOPIA                  |     3
 EGYPT                     |     6
 IRAN                      |     2
 SAUDI ARABIA              |     1
 RUSSIA                    |     5
 VIETNAM                   |     6
 PERU                      |     4
 ALGERIA                   |     3
 ARGENTINA                 |     3
 JAPAN                     |     4
 IRAQ                      |     2
 CANADA                    |     3
 BRAZIL                    |     2
 INDONESIA                 |     5
 FRANCE                    |     2
 JORDAN                    |     1
 MOROCCO                   |     2
 UNITED STATES             |     8
 CHINA                     |     7
 GERMANY                   |     5
 ROMANIA                   |     5
 INDIA                     |     5
 UNITED KINGDOM            |     3
 MOZAMBIQUE                |     7
 KENYA                     |     6
(25 rows)

postgres=# 
```

实验三 SQL 的编写

1. 单表查询

1.1 查询 PART 表中所有品牌名称以 "BRAND" 开头且类型包含 "POLISHED" 的商品，按商品尺寸降序排列

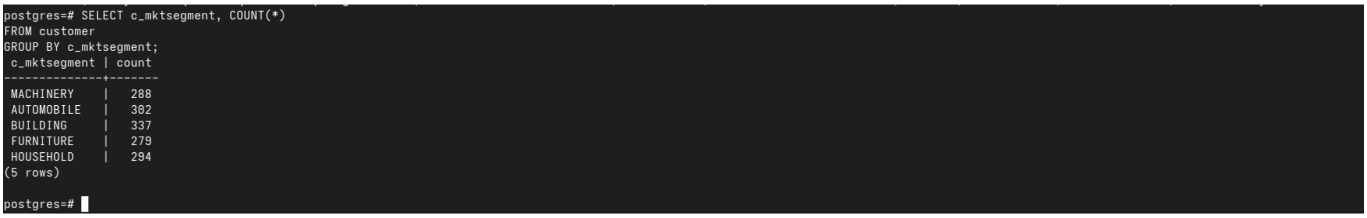
```
SELECT * FROM part
WHERE p_brand LIKE 'Brand%'
  AND p_type LIKE '%POLISHED%'
ORDER BY p_size DESC;
```



```
postgres=# SELECT * FROM part
WHERE p_brand LIKE 'Brand%'
  AND p_type LIKE '%POLISHED%'
ORDER BY p_size DESC;
 p_partkey |                         p_name                          |          p_mfgr           |  p_brand   |          p_type           | p_size | p_container | p_retailprice |        p_comment
-----------+---------------------------------------------------------+---------------------------+------------+---------------------------+--------+-------------+---------------+-------------------------
      1023 | grey blanched mint firebrick snow                       | Manufacturer#2            | Brand#24   | PROMO POLISHED COPPER     |     50 | SM CASE     |        924.02 | ideas use slyly. care
      1583 | cream deep floral grey chiffon                          | Manufacturer#1            | Brand#12   | ECONOMY POLISHED STEEL    |     50 | SM JAR      |       1484.58 |  cajole fluffily
       436 | turquoise yellow dim purple antique                     | Manufacturer#1            | Brand#14   | LARGE POLISHED BRASS      |     50 | WRAP CASE   |       1336.43 |  the regul
      1979 | tomato forest pale moccasin magenta                     | Manufacturer#5            | Brand#52   | SMALL POLISHED COPPER     |     50 | SM PKG      |       1880.97 | es. blithel
       843 | frosted metallic mint lawn blanched                     | Manufacturer#1            | Brand#15   | MEDIUM POLISHED STEEL     |     50 | LG PACK     |       1743.84 |  the
       767 | blush firebrick misty blanched purple                   | Manufacturer#2            | Brand#24   | LARGE POLISHED TIN        |     50 | MED DRUM    |       1667.76 | ts. carefully unu
      1333 | red sky dim light ivory                                 | Manufacturer#2            | Brand#23   | MEDIUM POLISHED STEEL     |     50 | WRAP JAR    |       1234.33 | ts alongside of the
      1572 | lemon midnight black papaya peru                        | Manufacturer#4            | Brand#45   | ECONOMY POLISHED COPPER   |     50 | LG JAR      |       1473.57 | riously ironic pinto
      1753 | beige smoke slate steel lavender                        | Manufacturer#2            | Brand#24   | ECONOMY POLISHED TIN      |     50 | MED PACK    |       1654.75 | y ironic bra
        90 | hot rosy violet plum pale                               | Manufacturer#5            | Brand#51   | ECONOMY POLISHED STEEL    |     49 | JUMBO CAN   |        990.09 | caref
        97 | coral dodger beige black chartreuse                     | Manufacturer#3            | Brand#33   | MEDIUM POLISHED BRASS     |     49 | WRAP CAN    |        997.09 | ss excuses sleep am
```

1.2 统计 CUSTOMER 表中每个市场区域 (MKTSEGMENT) 的客户数量

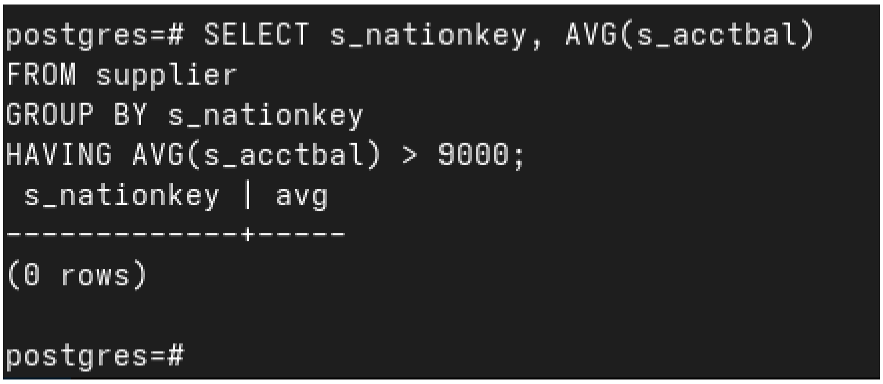
```
SELECT c_mktsegment, COUNT(*)
FROM customer
GROUP BY c_mktsegment;
```



```
postgres=# SELECT c_mktsegment, COUNT(*)
FROM customer
GROUP BY c_mktsegment;
 c_mktsegment | count
--------------+-------
 MACHINERY    |   288
 AUTOMOBILE   |   302
 BUILDING     |   337
 FURNITURE    |   279
 HOUSEHOLD    |   294
(5 rows)

postgres=#
```

1.3 查询 SUPPLIER 表中，账户余额 (ACCTBAL) 平均值大于 9000 的国家 (NATIONKEY) 及其平均余额

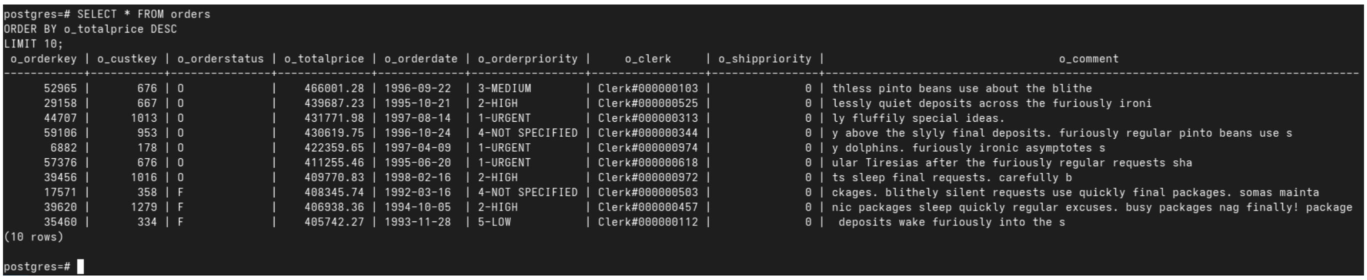
```
SELECT s_nationkey, AVG(s_acctbal)
FROM supplier
GROUP BY s_nationkey
HAVING AVG(s_acctbal) > 9000;
```



```
postgres=# SELECT s_nationkey, AVG(s_acctbal)
FROM supplier
GROUP BY s_nationkey
HAVING AVG(s_acctbal) > 9000;
 s_nationkey | avg
-------------+-----
(0 rows)

postgres=#
```

1.4 查询 ORDERS 表中订单总金额 (TOTALPRICE) 最高的前 10 笔订单信息

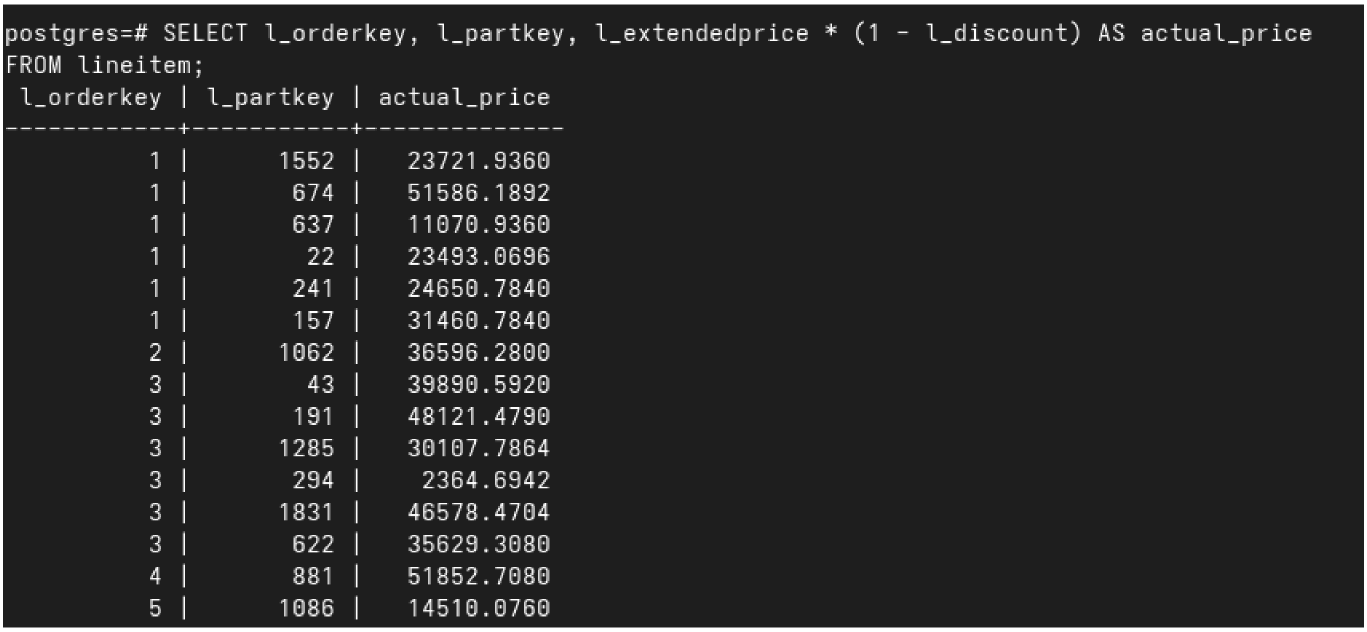
```
SELECT * FROM orders
ORDER BY o_totalprice DESC
LIMIT 10;
```



```
postgres=# SELECT * FROM orders
ORDER BY o_totalprice DESC
LIMIT 10;
 o_orderkey | o_custkey | o_orderstatus | o_totalprice | o_orderdate | o_orderpriority |     o_clerk     | o_shippriority |                                     o_comment
------------+-----------+---------------+--------------+-------------+-----------------+-----------------+----------------+-------------------------------------------------------------------------------
      52965 |       676 | O             |    466001.28 | 1996-09-22  | 3-MEDIUM        | Clerk#000000103 |              0 | thless pinto beans use about the blithe
      29158 |       667 | O             |    439687.23 | 1995-10-21  | 2-HIGH          | Clerk#000000525 |              0 | lessly quiet deposits across the furiously ironi
      44707 |      1013 | O             |    431771.98 | 1997-08-14  | 1-URGENT        | Clerk#000000313 |              0 | ly fluffily special ideas.
      59106 |       953 | O             |    430619.75 | 1996-10-24  | 4-NOT SPECIFIED | Clerk#000000344 |              0 | y above the slyly final deposits. furiously regular pinto beans use s
       6882 |       178 | O             |    422359.65 | 1997-04-09  | 1-URGENT        | Clerk#000000974 |              0 | y dolphins. furiously ironic asymptotes s
      57376 |       676 | O             |    411255.46 | 1995-06-20  | 1-URGENT        | Clerk#000000618 |              0 | ular Tiresias after the furiously regular requests sha
      39456 |      1016 | O             |    409770.83 | 1998-02-16  | 2-HIGH          | Clerk#000000972 |              0 | ts sleep final requests. carefully b
      17571 |       358 | F             |    408345.74 | 1992-03-16  | 4-NOT SPECIFIED | Clerk#000000503 |              0 | ckages. blithely silent requests use quickly final packages. somas mainta
      39620 |      1279 | F             |    406938.36 | 1994-10-05  | 2-HIGH          | Clerk#000000457 |              0 | nic packages sleep quickly regular excuses. busy packages nag finally! package
      35460 |       334 | F             |    405742.27 | 1993-11-28  | 5-LOW           | Clerk#000000112 |              0 |  deposits wake furiously into the s
(10 rows)

postgres=#
```

1.5 查询 LINEITEM 表中每条订单明细的实际金额（扩展价格 × (1 - 折扣)），并显示订单号、商品号和实际金额

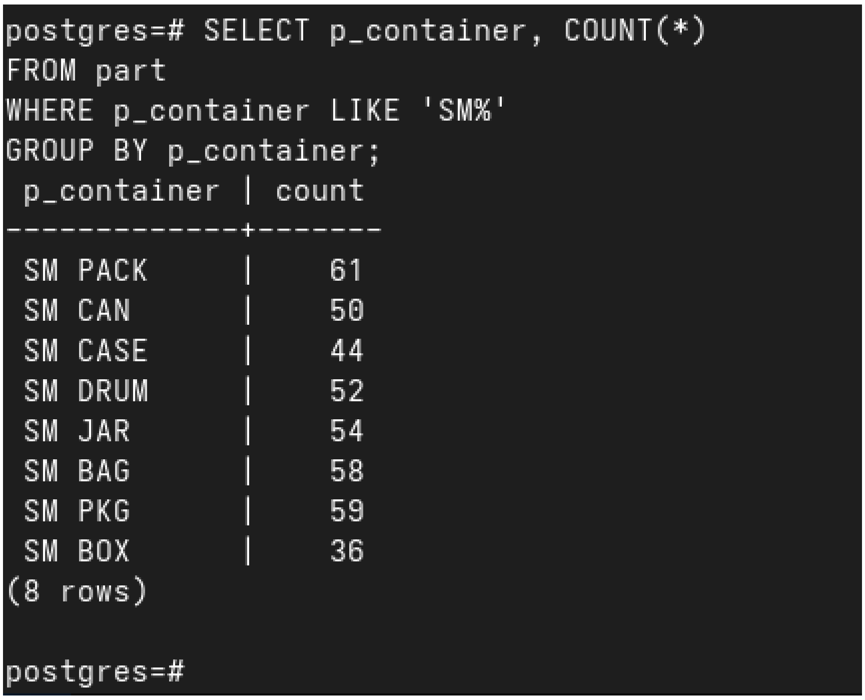
```
SELECT l_orderkey, l_partkey, l_extendedprice * (1 - l_discount) AS
actual_price
FROM lineitem;
```



```
postgres=# SELECT l_orderkey, l_partkey, l_extendedprice * (1 - l_discount) AS actual_price
FROM lineitem;
 l_orderkey | l_partkey | actual_price
------------+-----------+--------------
          1 |      1552 |   23721.9360
          1 |       674 |   51586.1892
          1 |       637 |   11070.9360
          1 |        22 |   23493.0696
          1 |       241 |   24650.7840
          1 |       157 |   31460.7840
          2 |      1062 |   36596.2800
          3 |        43 |   39890.5920
          3 |       191 |   48121.4790
          3 |      1285 |   30107.7864
          3 |       294 |    2364.6942
          3 |      1831 |   46578.4704
          3 |       622 |   35629.3080
          4 |       881 |   51852.7080
          5 |      1086 |   14510.0760
```

1.6 统计 PART 表中包装类型 (CONTAINER) 以 "SM" 开头的各类包装的商品数量

```sql
SELECT p_container, COUNT(*)
FROM part
WHERE p_container LIKE 'SM%'
GROUP BY p_container;
```

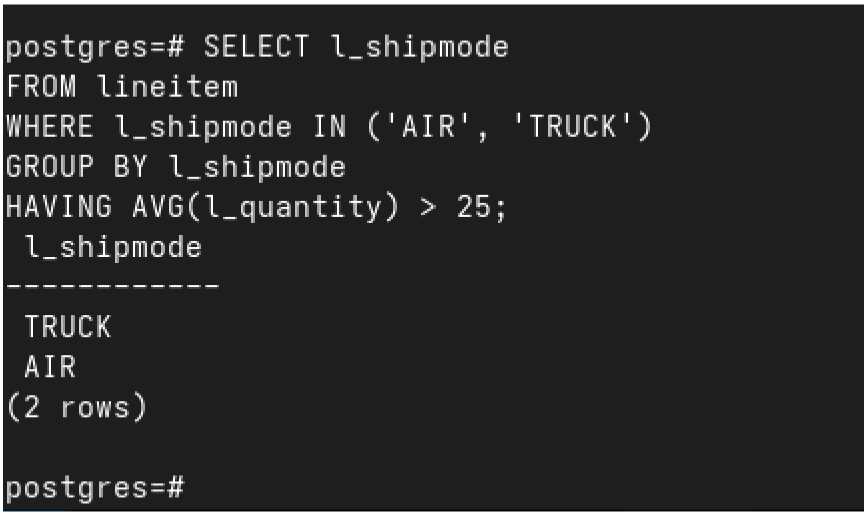


```
postgres=# SELECT p_container, COUNT(*)
FROM part
WHERE p_container LIKE 'SM%'
GROUP BY p_container;
 p_container | count
-------------+-------
 SM PACK     |    61
 SM CAN      |    50
 SM CASE     |    44
 SM DRUM     |    52
 SM JAR      |    54
 SM BAG      |    58
 SM PKG      |    59
 SM BOX      |    36
(8 rows)

postgres=#
```

1.7 查询 LINEITEM 表中，发货模式 (SHIPMODE) 为 "AIR" 或 "TRUCK"，且平均数量大于 25 的发货模式

```sql
SELECT l_shipmode
FROM lineitem
WHERE l_shipmode IN ('AIR', 'TRUCK')
```

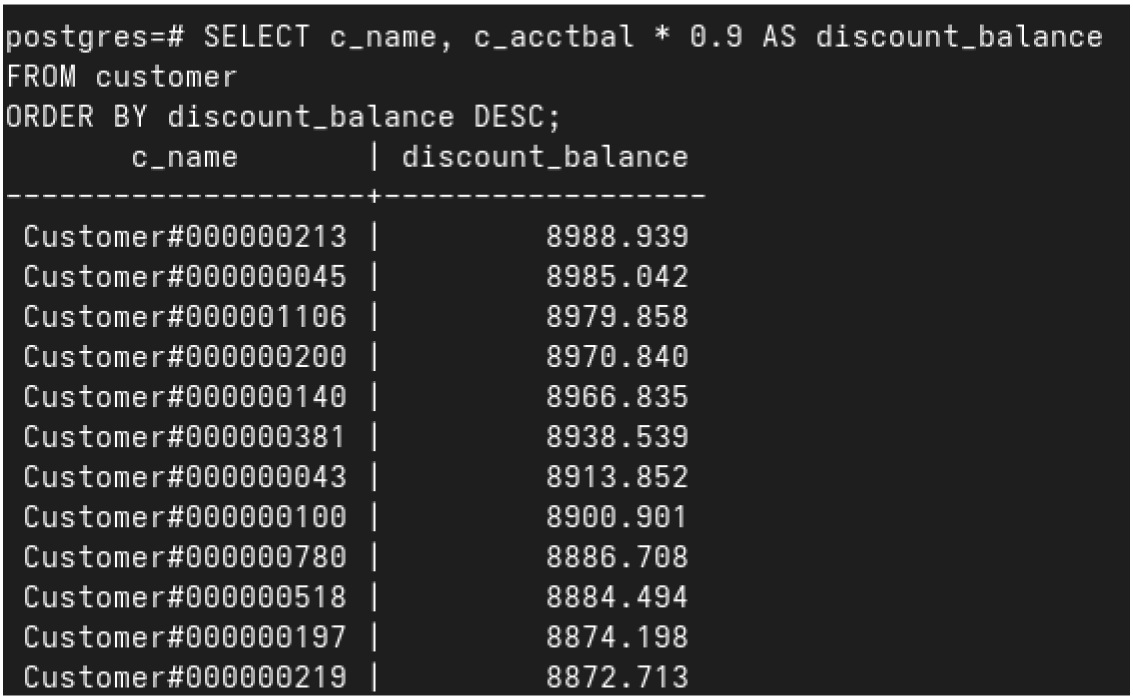
```
GROUP BY l_shipmode
HAVING AVG(l_quantity) > 25;
```



```
postgres=# SELECT l_shipmode
FROM lineitem
WHERE l_shipmode IN ('AIR', 'TRUCK')
GROUP BY l_shipmode
HAVING AVG(l_quantity) > 25;
 l_shipmode
------------
 TRUCK
 AIR
(2 rows)

postgres=#
```

1.8 查询 CUSTOMER 表，计算每个客户的折扣余额（账户余额 × 0.9），并按折扣余额降序排列

```
SELECT c_name, c_acctbal * 0.9 AS discount_balance
FROM customer
ORDER BY discount_balance DESC;
```

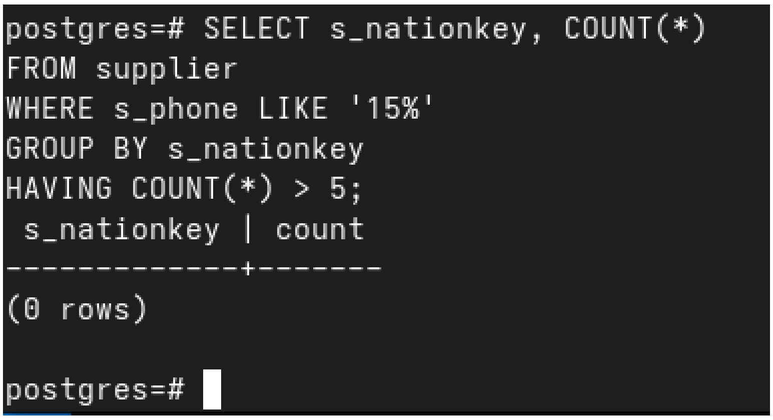


```
postgres=# SELECT c_name, c_acctbal * 0.9 AS discount_balance
FROM customer
ORDER BY discount_balance DESC;
       c_name       | discount_balance
--------------------+------------------
 Customer#000000213 |         8988.939
 Customer#000000045 |         8985.042
 Customer#000001106 |         8979.858
 Customer#000000200 |         8970.840
 Customer#000000140 |         8966.835
 Customer#000000381 |         8938.539
 Customer#000000043 |         8913.852
 Customer#000000100 |         8900.901
 Customer#000000780 |         8886.708
 Customer#000000518 |         8884.494
 Customer#000000197 |         8874.198
 Customer#000000219 |         8872.713
```

1.9 查询 SUPPLIER 表中，电话号码以 "15" 开头的供应商，按国家分组后，筛选出供应商数量超过 5 个的国家

```
SELECT s_nationkey, COUNT(*)
FROM supplier
WHERE s_phone LIKE '15%'
```

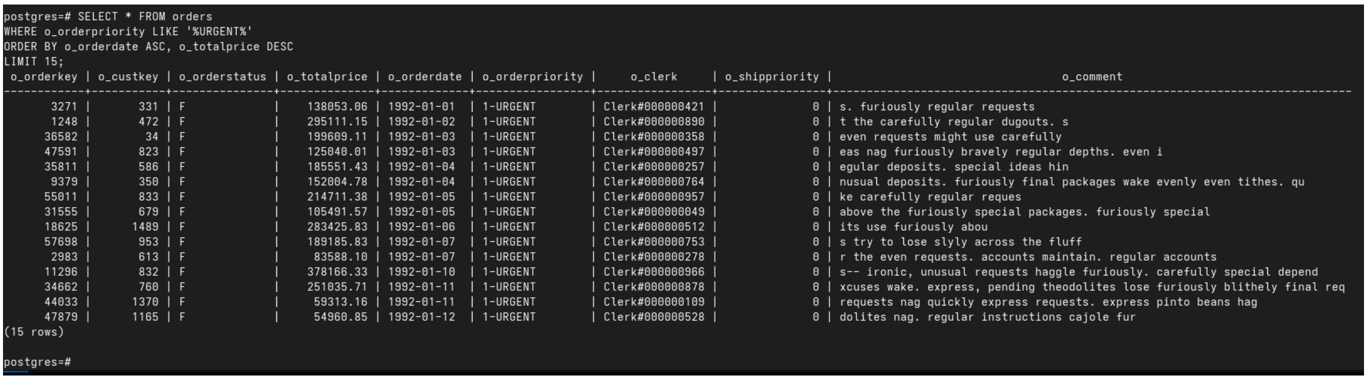
```
GROUP BY s_nationkey
HAVING COUNT(*) > 5;
```



```
postgres=# SELECT s_nationkey, COUNT(*)
FROM supplier
WHERE s_phone LIKE '15%'
GROUP BY s_nationkey
HAVING COUNT(*) > 5;
 s_nationkey | count
-------------+-------
(0 rows)

postgres=#
```

1.10 查询 ORDERS 表中订单优先级 (PRIORITY) 包含 "URGENT" 的订单，按订单日期升序、总金额降序排列，显示前 15 条

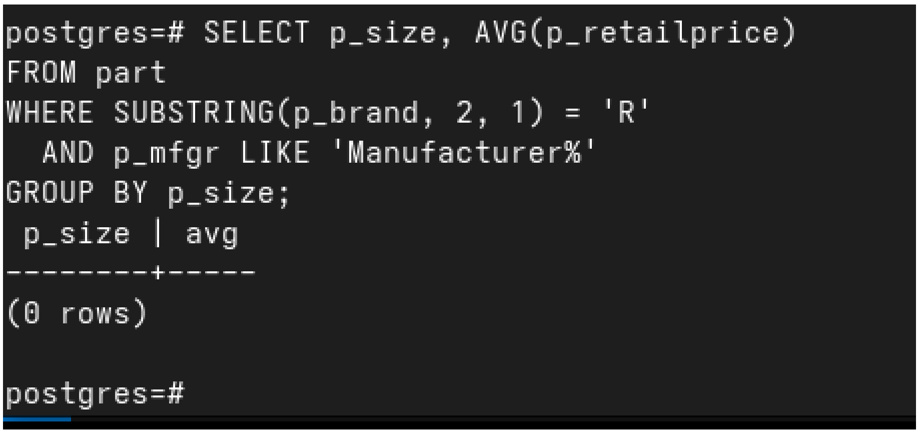
```
SELECT * FROM orders
WHERE o_orderpriority LIKE '%URGENT%'
ORDER BY o_orderdate ASC, o_totalprice DESC
LIMIT 15;
```



```
postgres=# SELECT * FROM orders
WHERE o_orderpriority LIKE '%URGENT%'
ORDER BY o_orderdate ASC, o_totalprice DESC
LIMIT 15;
 o_orderkey | o_custkey | o_orderstatus | o_totalprice | o_orderdate | o_orderpriority |     o_clerk     | o_shippriority |                                o_comment
------------+-----------+---------------+--------------+-------------+-----------------+-----------------+----------------+--------------------------------------------------------------------------
       3271 |       331 | F             |    138053.06 | 1992-01-01  | 1-URGENT        | Clerk#000000421 |              0 | s. furiously regular requests
       1248 |       472 | F             |    295111.15 | 1992-01-02  | 1-URGENT        | Clerk#000000890 |              0 | t the carefully regular dugouts. s
      36582 |        34 | F             |    199609.11 | 1992-01-03  | 1-URGENT        | Clerk#000000358 |              0 | even requests might use carefully
      47591 |       823 | F             |    125040.01 | 1992-01-03  | 1-URGENT        | Clerk#000000497 |              0 | eas nag furiously bravely regular depths. even i
      35811 |       586 | F             |    185551.43 | 1992-01-04  | 1-URGENT        | Clerk#000000257 |              0 | egular deposits. special ideas hin
       9379 |       350 | F             |    152004.78 | 1992-01-04  | 1-URGENT        | Clerk#000000764 |              0 | nusual deposits. furiously final packages wake evenly even tithes. qu
      55011 |       833 | F             |    214711.38 | 1992-01-05  | 1-URGENT        | Clerk#000000957 |              0 | ke carefully regular reques
      31555 |       679 | F             |    105491.57 | 1992-01-05  | 1-URGENT        | Clerk#000000049 |              0 | above the furiously special packages. furiously special
      18625 |      1489 | F             |    283425.83 | 1992-01-06  | 1-URGENT        | Clerk#000000512 |              0 | its use furiously abou
      57698 |       953 | F             |    189185.83 | 1992-01-07  | 1-URGENT        | Clerk#000000753 |              0 | s try to lose slyly across the fluff
       2983 |       613 | F             |     83588.10 | 1992-01-07  | 1-URGENT        | Clerk#000000278 |              0 | r the even requests. accounts maintain. regular accounts
      11296 |       832 | F             |    378166.33 | 1992-01-10  | 1-URGENT        | Clerk#000000966 |              0 | s-- ironic, unusual requests haggle furiously. carefully special depend
      34662 |       760 | F             |    251035.71 | 1992-01-11  | 1-URGENT        | Clerk#000000878 |              0 | xcuses wake. express, pending theodolites lose furiously blithely final req
      44033 |      1370 | F             |     59313.16 | 1992-01-11  | 1-URGENT        | Clerk#000000109 |              0 | requests nag quickly express requests. express pinto beans hag
      47879 |      1165 | F             |     54960.85 | 1992-01-12  | 1-URGENT        | Clerk#000000528 |              0 | dolites nag. regular instructions cajole fur
(15 rows)

postgres=#
```

1.11 查询 PART 表，找出品牌名称第二个字符为大写字母 "R"，且制造商名称以 "Manufacturer" 开头的商品，计算每个尺寸组的平均零售价

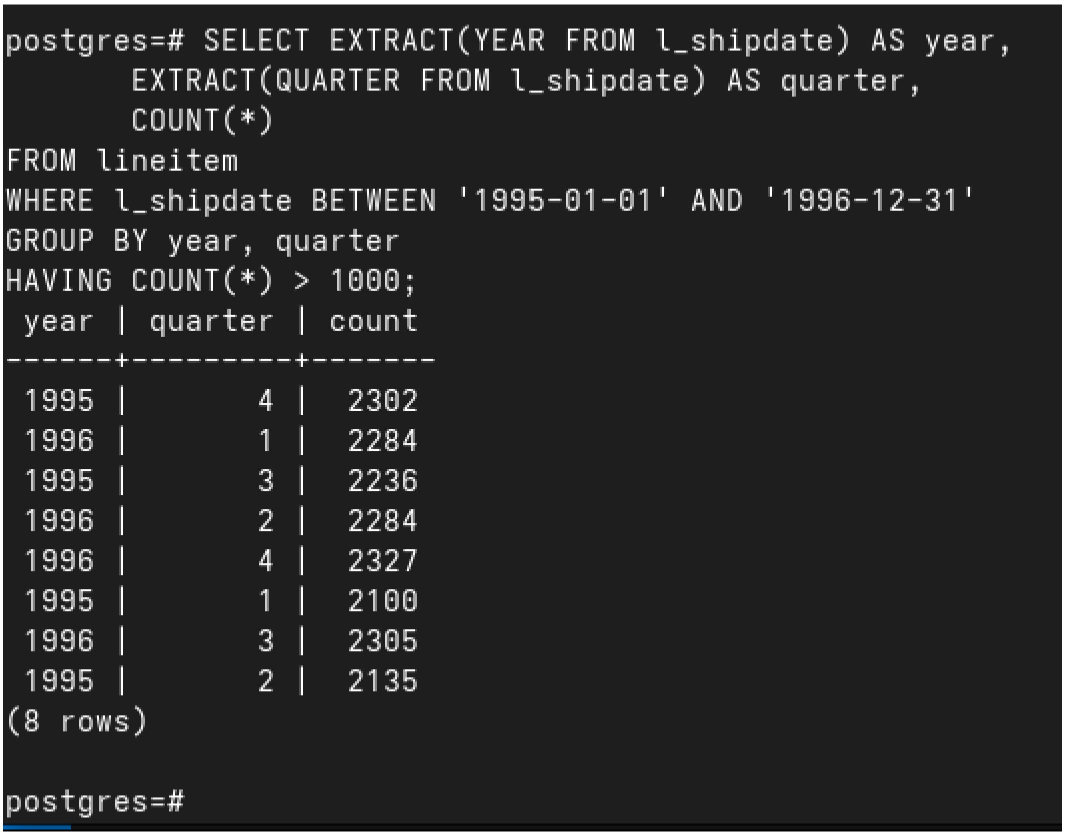
```
SELECT p_size, AVG(p_retailprice)
FROM part
WHERE SUBSTRING(p_brand, 2, 1) = 'R'
  AND p_mfgr LIKE 'Manufacturer%'
GROUP BY p_size;
```



```
postgres=# SELECT p_size, AVG(p_retailprice)
FROM part
WHERE SUBSTRING(p_brand, 2, 1) = 'R'
  AND p_mfgr LIKE 'Manufacturer%'
GROUP BY p_size;
 p_size | avg
--------+-----
(0 rows)

postgres=#
```

1.12 查询 LINEITEM 表，统计 1995-1996 年间每个季度的订单明细数量，只显示数量超过 1000 的季度

```
SELECT EXTRACT(YEAR FROM l_shipdate) AS year,
       EXTRACT(QUARTER FROM l_shipdate) AS quarter,
       COUNT(*)
FROM lineitem
WHERE l_shipdate BETWEEN '1995-01-01' AND '1996-12-31'
GROUP BY year, quarter
HAVING COUNT(*) > 1000;
```

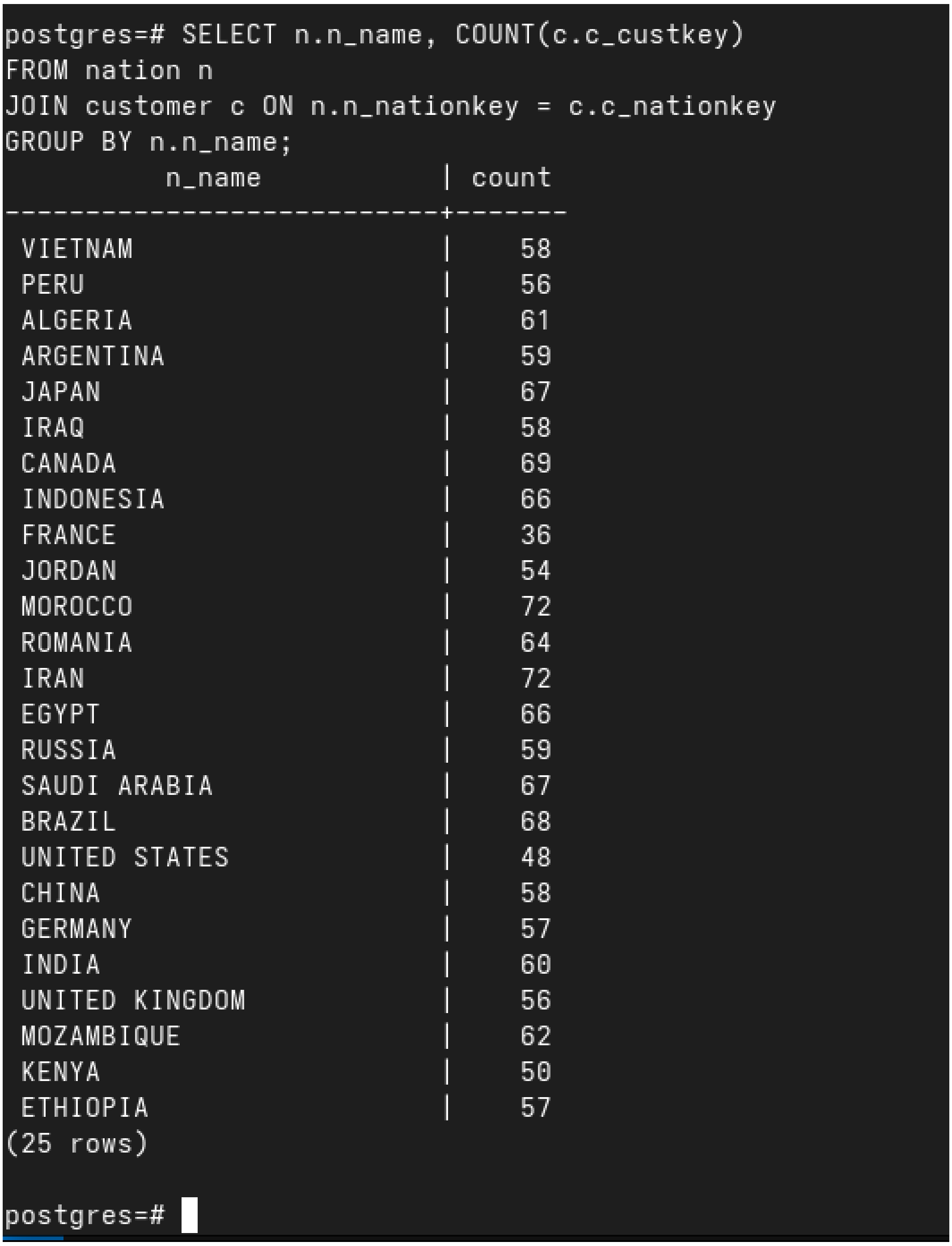


```
postgres=# SELECT EXTRACT(YEAR FROM l_shipdate) AS year,
       EXTRACT(QUARTER FROM l_shipdate) AS quarter,
       COUNT(*)
FROM lineitem
WHERE l_shipdate BETWEEN '1995-01-01' AND '1996-12-31'
GROUP BY year, quarter
HAVING COUNT(*) > 1000;
 year | quarter | count
------+---------+-------
 1995 |       4 |  2302
 1996 |       1 |  2284
 1995 |       3 |  2236
 1996 |       2 |  2284
 1996 |       4 |  2327
 1995 |       1 |  2100
 1996 |       3 |  2305
 1995 |       2 |  2135
(8 rows)

postgres=#
```

2. 多表查询

2.1 查询每个国家 (NATION) 的客户数量，显示国家名称和客户数量

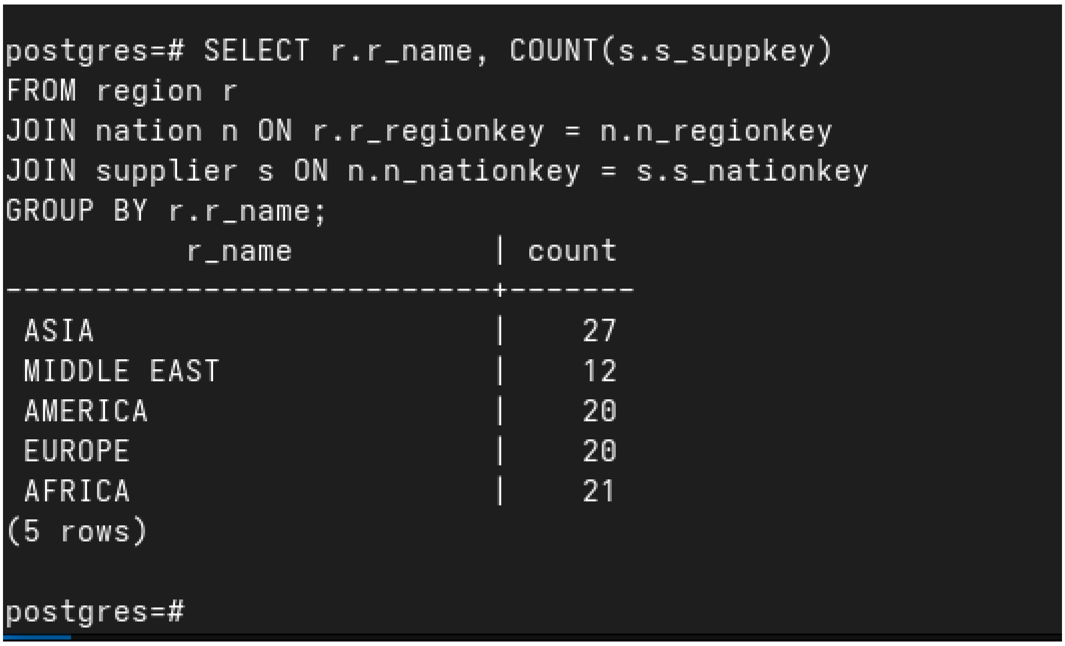
```
SELECT n.n_name, COUNT(c.c_custkey)
FROM nation n
JOIN customer c ON n.n_nationkey = c.c_nationkey
GROUP BY n.n_name;
```



```
postgres=# SELECT n.n_name, COUNT(c.c_custkey)
FROM nation n
JOIN customer c ON n.n_nationkey = c.c_nationkey
GROUP BY n.n_name;
          n_name           | count
---------------------------+-------
 VIETNAM                   |    58
 PERU                      |    56
 ALGERIA                   |    61
 ARGENTINA                 |    59
 JAPAN                     |    67
 IRAQ                      |    58
 CANADA                    |    69
 INDONESIA                 |    66
 FRANCE                    |    36
 JORDAN                    |    54
 MOROCCO                   |    72
 ROMANIA                   |    64
 IRAN                      |    72
 EGYPT                     |    66
 RUSSIA                    |    59
 SAUDI ARABIA              |    67
 BRAZIL                    |    68
 UNITED STATES             |    48
 CHINA                     |    58
 GERMANY                   |    57
 INDIA                     |    60
 UNITED KINGDOM            |    56
 MOZAMBIQUE                |    62
 KENYA                     |    50
 ETHIOPIA                  |    57
(25 rows)

postgres=#
```

2.2 查询每个地区 (REGION) 的供应商数量，显示地区名称和供应商数量

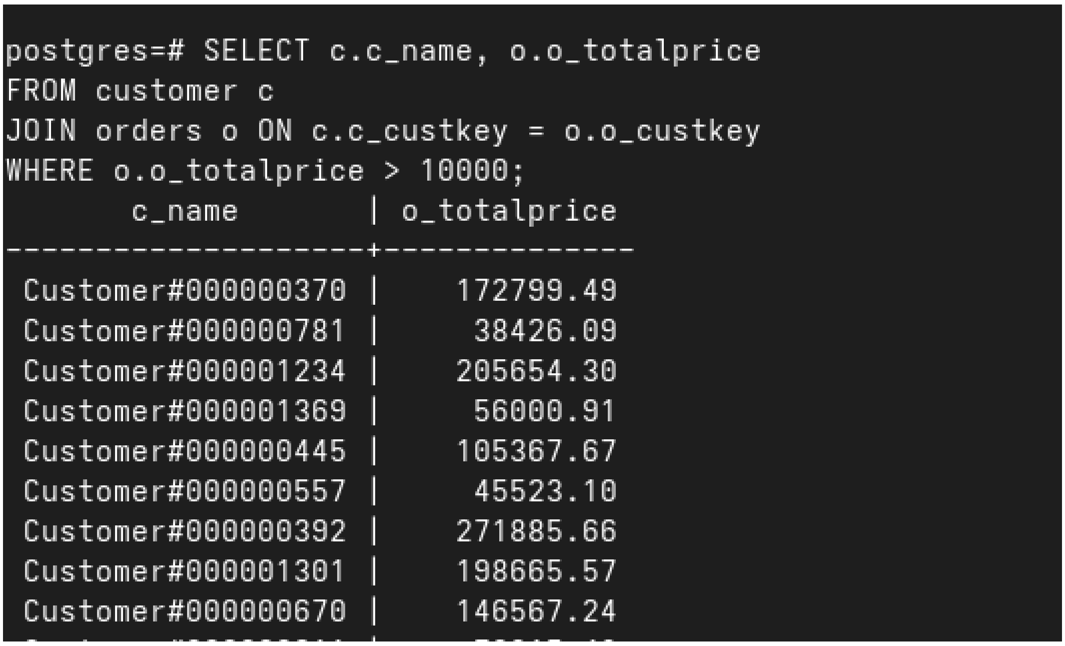
```
SELECT r.r_name, COUNT(s.s_suppkey)
FROM region r
JOIN nation n ON r.r_regionkey = n.n_regionkey
JOIN supplier s ON n.n_nationkey = s.s_nationkey
GROUP BY r.r_name;
```



```
postgres=# SELECT r.r_name, COUNT(s.s_suppkey)
FROM region r
JOIN nation n ON r.r_regionkey = n.n_regionkey
JOIN supplier s ON n.n_nationkey = s.s_nationkey
GROUP BY r.r_name;
          r_name           | count
---------------------------+-------
 ASIA                      |    27
 MIDDLE EAST               |    12
 AMERICA                   |    20
 EUROPE                    |    20
 AFRICA                    |    21
(5 rows)

postgres=#
```

2.3 查询单个订单总金额超过 10000 的客户名称及其订单总金额

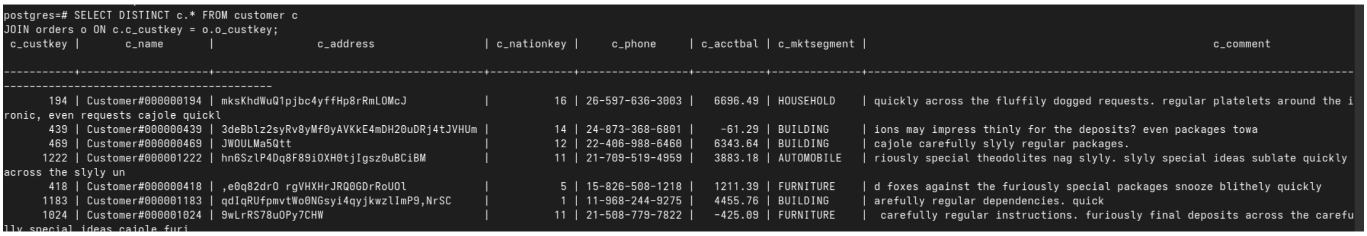
```sql
SELECT c.c_name, o.o_totalprice
FROM customer c
JOIN orders o ON c.c_custkey = o.o_custkey
WHERE o.o_totalprice > 10000;
```



```
postgres=# SELECT c.c_name, o.o_totalprice
FROM customer c
JOIN orders o ON c.c_custkey = o.o_custkey
WHERE o.o_totalprice > 10000;
       c_name       | o_totalprice
--------------------+--------------
 Customer#000000370 |    172799.49
 Customer#000000781 |     38426.09
 Customer#000001234 |    205654.30
 Customer#000001369 |     56000.91
 Customer#000000445 |    105367.67
 Customer#000000557 |     45523.10
 Customer#000000392 |    271885.66
 Customer#000001301 |    198665.57
 Customer#000000670 |    146567.24
```

2.4 查询至少有一个订单的客户信息

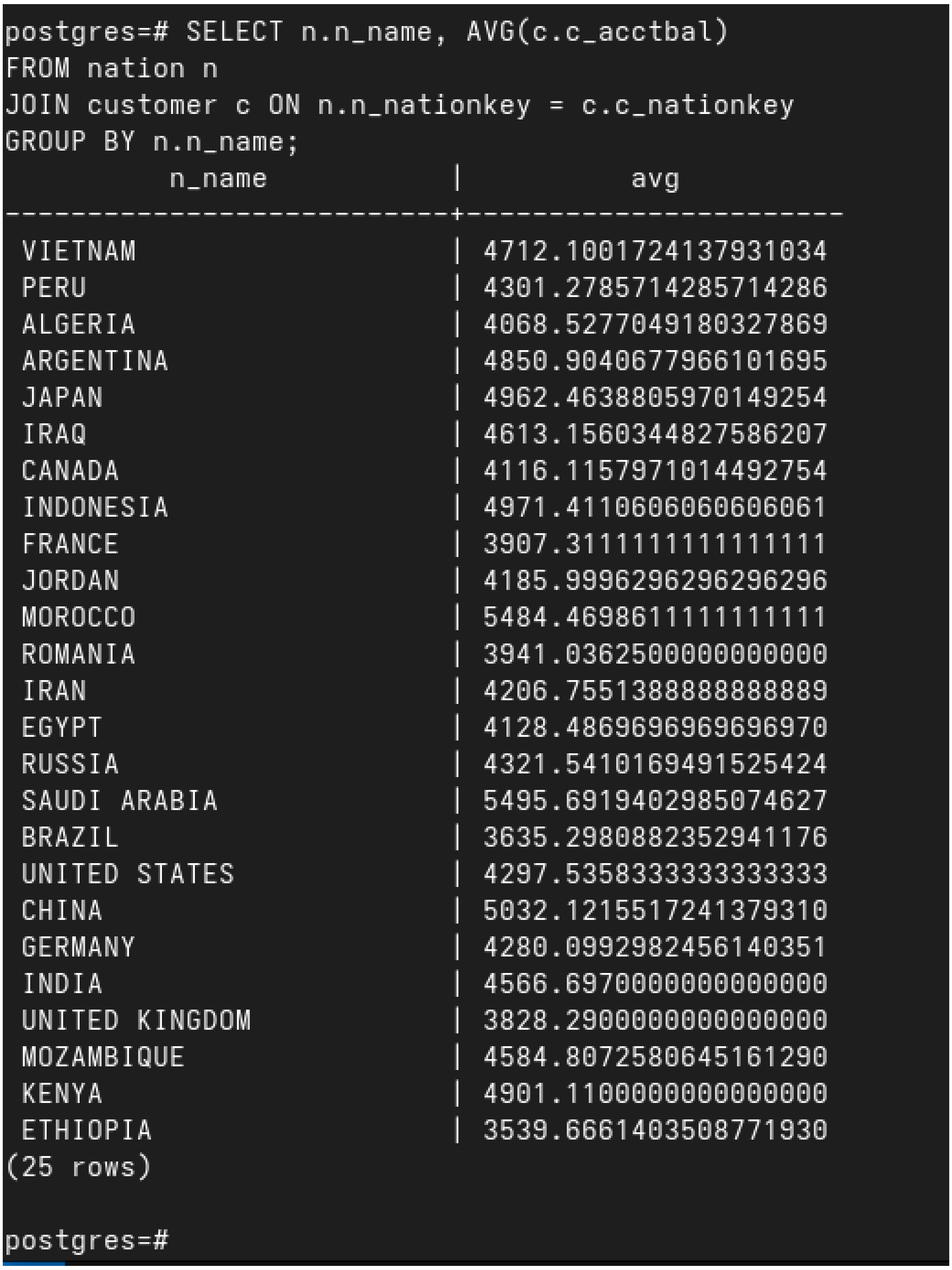
```sql
SELECT DISTINCT c.* FROM customer c
JOIN orders o ON c.c_custkey = o.o_custkey;
```



```
postgres=# SELECT DISTINCT c.* FROM customer c
JOIN orders o ON c.c_custkey = o.o_custkey;
 c_custkey |       c_name       |                 c_address                 | c_nationkey |      c_phone      | c_acctbal | c_mktsegment |                                          c_comment

-----------+--------------------+-------------------------------------------+-------------+-------------------+-----------+--------------+--------------------------------------------------------------------------------------
----------------------------------------
       194 | Customer#000000194 | mksKhdWuQ1pjbc4yffHp8rRmLOMcJ             |          16 | 26-597-636-3003 |   6696.49 | HOUSEHOLD    | quickly across the fluffily dogged requests. regular platelets around the i
ronic, even requests cajole quickl
       439 | Customer#000000439 | 3deBblz2syRv8yMf0yAVKkE4mDH20uDRj4tJVHUm |          14 | 24-873-368-6801 |    -61.29 | BUILDING     | ions may impress thinly for the deposits? even packages towa
       469 | Customer#000000469 | JWOULMa5Qtt                               |          12 | 22-406-988-6460 |   6343.64 | BUILDING     | cajole carefully slyly regular packages.
      1222 | Customer#000001222 | hn6SzlP4Dq8F89iOXH0tjIgsz0uBCiBM          |          11 | 21-709-519-4959 |   3883.18 | AUTOMOBILE   | riously special theodolites nag slyly. slyly special ideas sublate quickly
across the slyly un
       418 | Customer#000000418 | ,e0q82dr0 rgVHXHrJRQ0GDrRoU0l             |           5 | 15-826-508-1218 |   1211.39 | FURNITURE    | d foxes against the furiously special packages snooze blithely quickly
      1183 | Customer#000001183 | qdIqRUfpmvtWo0NGsyi4qyjkwzlImP9,NrSC      |           1 | 11-968-244-9275 |   4455.76 | BUILDING     | arefully regular dependencies. quick
      1024 | Customer#000001024 | 9wLrRS78uOPy7CHW                          |          11 | 21-508-779-7822 |   -425.09 | FURNITURE    |  carefully regular instructions. furiously final deposits across the carefu
lly special ideas cajole furi
```

2.5 查询每个国家的平均客户账户余额，结果显示国家名称和平均客户账户余额

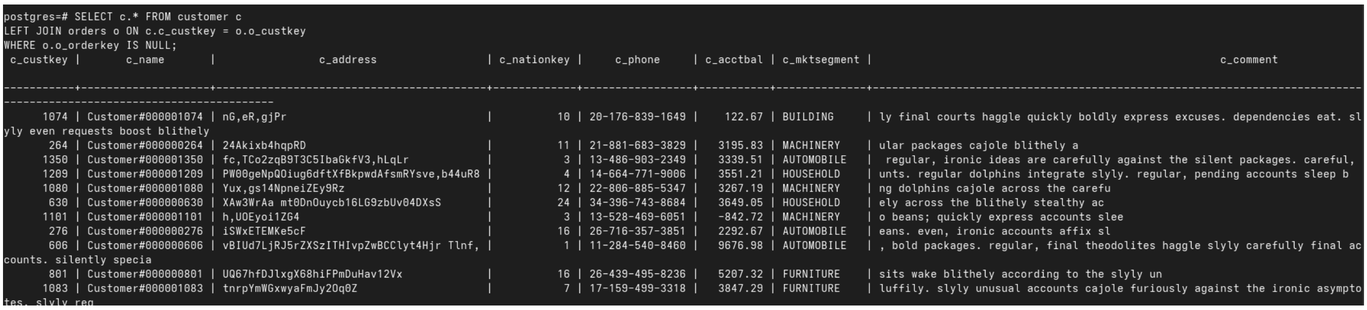
```sql
SELECT n.n_name, AVG(c.c_acctbal)
FROM nation n
JOIN customer c ON n.n_nationkey = c.c_nationkey
GROUP BY n.n_name;
```



```
postgres=# SELECT n.n_name, AVG(c.c_acctbal)
FROM nation n
JOIN customer c ON n.n_nationkey = c.c_nationkey
GROUP BY n.n_name;
          n_name           |          avg
---------------------------+-----------------------
 VIETNAM                   | 4712.1001724137931034
 PERU                      | 4301.2785714285714286
 ALGERIA                   | 4068.5277049180327869
 ARGENTINA                 | 4850.9040677966101695
 JAPAN                     | 4962.4638805970149254
 IRAQ                      | 4613.1560344827586207
 CANADA                    | 4116.1157971014492754
 INDONESIA                 | 4971.4110606060606061
 FRANCE                    | 3907.3111111111111111
 JORDAN                    | 4185.9996296296296296
 MOROCCO                   | 5484.4698611111111111
 ROMANIA                   | 3941.0362500000000000
 IRAN                      | 4206.7551388888888889
 EGYPT                     | 4128.4869696969696970
 RUSSIA                    | 4321.5410169491525424
 SAUDI ARABIA              | 5495.6919402985074627
 BRAZIL                    | 3635.2980882352941176
 UNITED STATES             | 4297.5358333333333333
 CHINA                     | 5032.1215517241379310
 GERMANY                   | 4280.0992982456140351
 INDIA                     | 4566.6970000000000000
 UNITED KINGDOM            | 3828.2900000000000000
 MOZAMBIQUE                | 4584.8072580645161290
 KENYA                     | 4901.1100000000000000
 ETHIOPIA                  | 3539.6661403508771930
(25 rows)

postgres=#
```

2.6 查询没有下过任何订单的客户信息

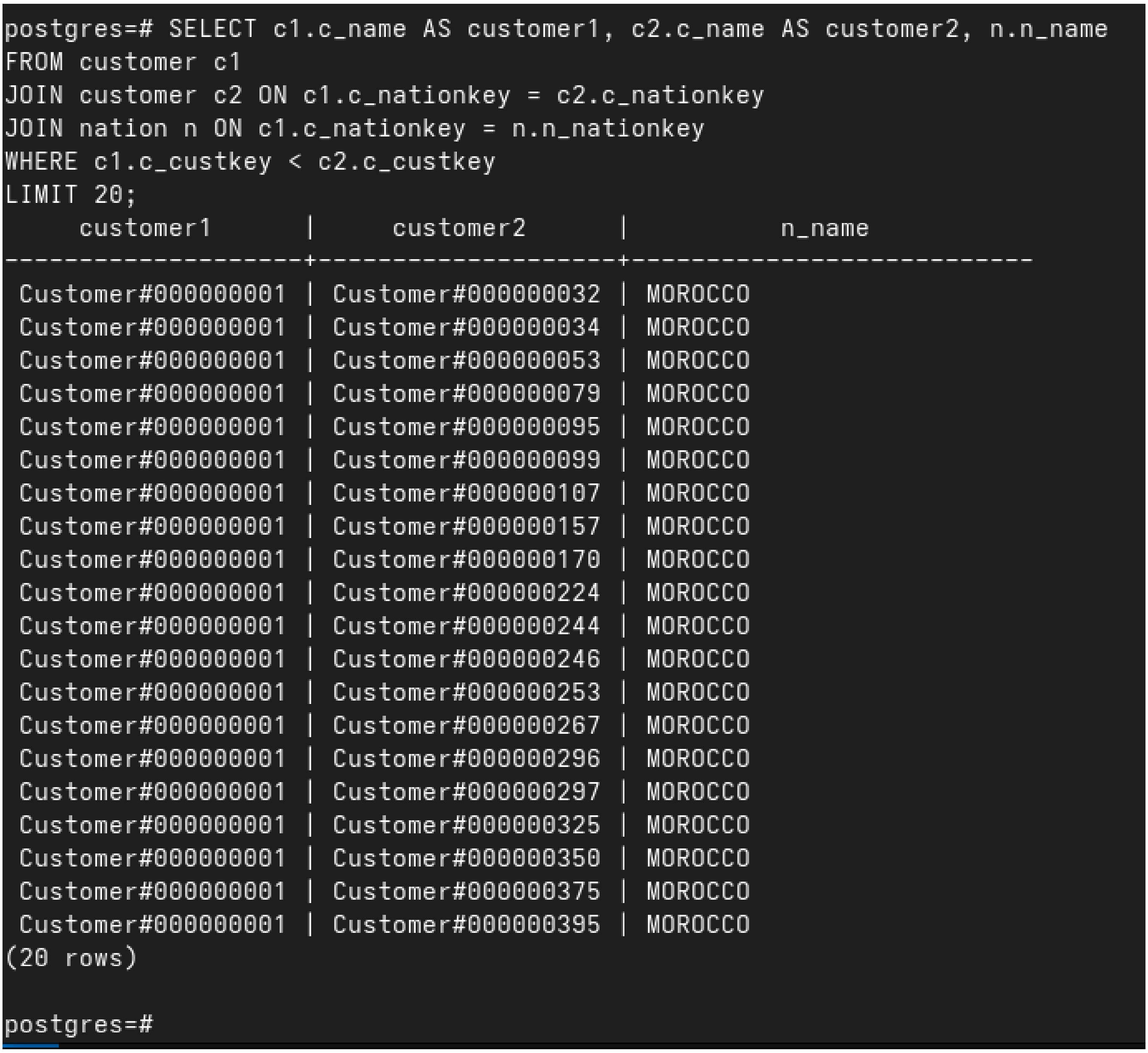
```sql
SELECT c.* FROM customer c
LEFT JOIN orders o ON c.c_custkey = o.o_custkey
WHERE o.o_orderkey IS NULL;
```



```
postgres=# SELECT c.* FROM customer c
LEFT JOIN orders o ON c.c_custkey = o.o_custkey
WHERE o.o_orderkey IS NULL;
 c_custkey |       c_name       |                 c_address                | c_nationkey |     c_phone     | c_acctbal | c_mktsegment |                                                     c_comment

-----------+--------------------+------------------------------------------+-------------+-----------------+-----------+--------------+----------------------------------------------------------------------
------------------------------------------
      1074 | Customer#000001074 | nG,eR,gjPr                               |          10 | 20-176-839-1649 |    122.67 | BUILDING     | ly final courts haggle quickly boldly express excuses. dependencies eat. sl
yly even requests boost blithely
       264 | Customer#000000264 | 24Akixb4hqpRD                            |          11 | 21-881-683-3829 |   3195.83 | MACHINERY    | ular packages cajole blithely a
      1350 | Customer#000001350 | fc,TCo2zqB9T3C5IbaGkfV3,hLqLr            |           3 | 13-486-903-2349 |   3339.51 | AUTOMOBILE   |  regular, ironic ideas are carefully against the silent packages. careful,
      1209 | Customer#000001209 | PW00geNpQOiug6dftXfBkpwdAfsmRYsve,b44uR8 |           4 | 14-664-771-9006 |   3551.21 | HOUSEHOLD    | unts. regular dolphins integrate slyly. regular, pending accounts sleep b
      1080 | Customer#000001080 | Yux,gs14NpneiZEy9Rz                      |          12 | 22-806-885-5347 |   3267.19 | MACHINERY    | ng dolphins cajole across the carefu
       630 | Customer#000000630 | XAw3WrAa mt0DnOuycb16LG9zbUv04DXsS       |          24 | 34-396-743-8684 |   3649.05 | HOUSEHOLD    | ely across the blithely stealthy ac
      1101 | Customer#000001101 | h,UOEyoi1ZG4                             |           3 | 13-528-469-6051 |   -842.72 | MACHINERY    | o beans; quickly express accounts slee
       276 | Customer#000000276 | iSWxETEMKe5cF                            |          16 | 26-716-357-3851 |   2292.67 | AUTOMOBILE   | eans. even, ironic accounts affix sl
       606 | Customer#000000606 | vBIUd7LjRJ5rZXSzITHIvpZwBCClyt4Hjr Tlnf, |           1 | 11-284-540-8460 |   9676.98 | AUTOMOBILE   | , bold packages. regular, final theodolites haggle slyly carefully final ac
counts. silently specia
       801 | Customer#000000801 | UQ67hfDJlxgX68hiFPmDuHav12Vx             |          16 | 26-439-495-8236 |   5207.32 | FURNITURE    | sits wake blithely according to the slyly un
      1083 | Customer#000001083 | tnrpYmWGxwyaFmJy20q0Z                    |           7 | 17-159-499-3318 |   3847.29 | FURNITURE    | luffily. slyly unusual accounts cajole furiously against the ironic asympto
tes. slyly reg
```

2.7 查询在同一国家的客户对（不重复组合）

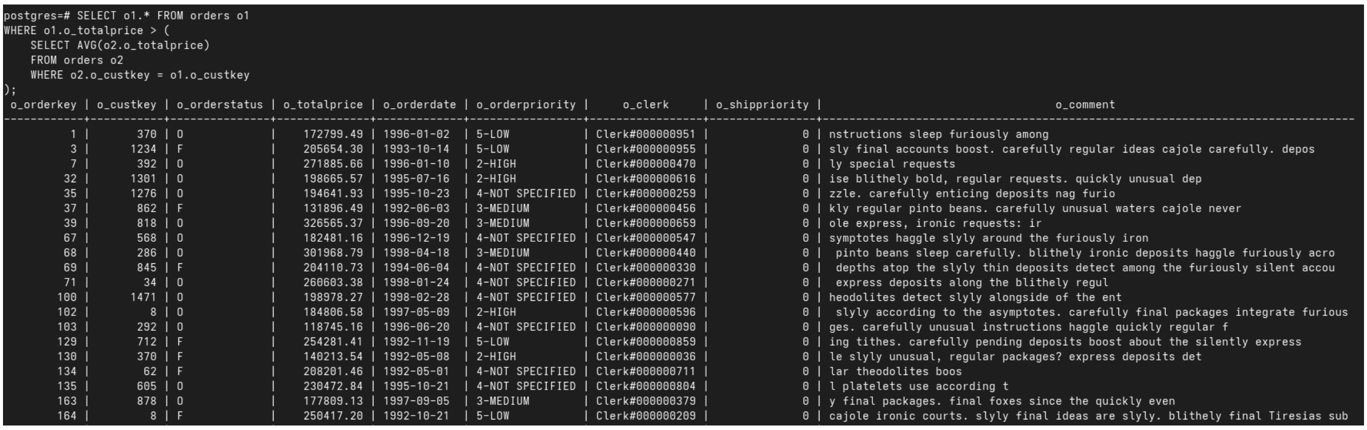
```sql
SELECT c1.c_name AS customer1, c2.c_name AS customer2, n.n_name
FROM customer c1
JOIN customer c2 ON c1.c_nationkey = c2.c_nationkey
JOIN nation n ON c1.c_nationkey = n.n_nationkey
WHERE c1.c_custkey < c2.c_custkey
LIMIT 20;
```



```
postgres=# SELECT c1.c_name AS customer1, c2.c_name AS customer2, n.n_name
FROM customer c1
JOIN customer c2 ON c1.c_nationkey = c2.c_nationkey
JOIN nation n ON c1.c_nationkey = n.n_nationkey
WHERE c1.c_custkey < c2.c_custkey
LIMIT 20;
     customer1      |     customer2      |          n_name
--------------------+--------------------+---------------------------
 Customer#000000001 | Customer#000000032 | MOROCCO
 Customer#000000001 | Customer#000000034 | MOROCCO
 Customer#000000001 | Customer#000000053 | MOROCCO
 Customer#000000001 | Customer#000000079 | MOROCCO
 Customer#000000001 | Customer#000000095 | MOROCCO
 Customer#000000001 | Customer#000000099 | MOROCCO
 Customer#000000001 | Customer#000000107 | MOROCCO
 Customer#000000001 | Customer#000000157 | MOROCCO
 Customer#000000001 | Customer#000000170 | MOROCCO
 Customer#000000001 | Customer#000000224 | MOROCCO
 Customer#000000001 | Customer#000000244 | MOROCCO
 Customer#000000001 | Customer#000000246 | MOROCCO
 Customer#000000001 | Customer#000000253 | MOROCCO
 Customer#000000001 | Customer#000000267 | MOROCCO
 Customer#000000001 | Customer#000000296 | MOROCCO
 Customer#000000001 | Customer#000000297 | MOROCCO
 Customer#000000001 | Customer#000000325 | MOROCCO
 Customer#000000001 | Customer#000000350 | MOROCCO
 Customer#000000001 | Customer#000000375 | MOROCCO
 Customer#000000001 | Customer#000000395 | MOROCCO
(20 rows)

postgres=#
```

2.8 查询订单总金额超过该客户平均订单金额的订单信息

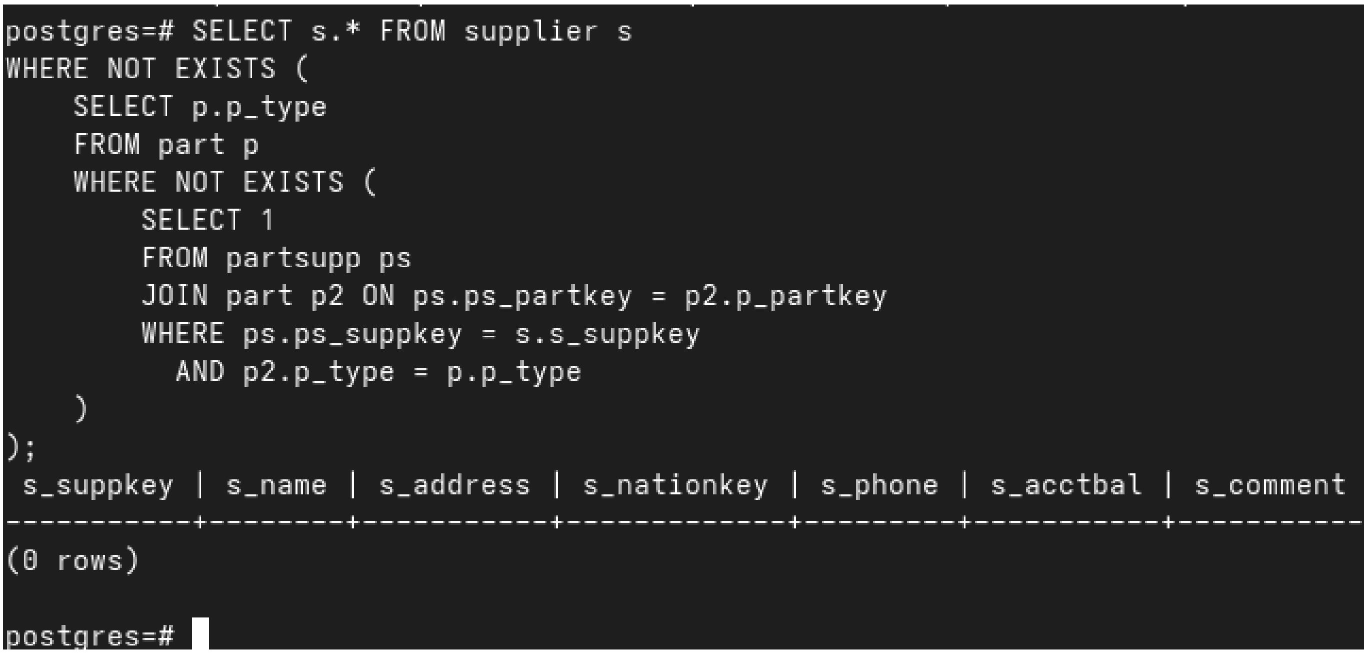
```sql
SELECT o1.* FROM orders o1
WHERE o1.o_totalprice > (
    SELECT AVG(o2.o_totalprice)
    FROM orders o2
    WHERE o2.o_custkey = o1.o_custkey
);
```



```
postgres=# SELECT o1.* FROM orders o1
WHERE o1.o_totalprice > (
    SELECT AVG(o2.o_totalprice)
    FROM orders o2
    WHERE o2.o_custkey = o1.o_custkey
);
 o_orderkey | o_custkey | o_orderstatus | o_totalprice | o_orderdate | o_orderpriority |     o_clerk     | o_shippriority |                                o_comment
------------+-----------+---------------+--------------+-------------+-----------------+-----------------+----------------+---------------------------------------------------------------------------
          1 |       370 | O             |    172799.49 | 1996-01-02  | 5-LOW           | Clerk#000000951 |              0 | nstructions sleep furiously among
          3 |      1234 | F             |    205654.30 | 1993-10-14  | 5-LOW           | Clerk#000000955 |              0 | sly final accounts boost. carefully regular ideas cajole carefully. depos
          7 |       392 | O             |    271885.66 | 1996-01-10  | 2-HIGH          | Clerk#000000470 |              0 | ly special requests
         32 |      1301 | O             |    198665.57 | 1995-07-16  | 2-HIGH          | Clerk#000000616 |              0 | ise blithely bold, regular requests. quickly unusual dep
         35 |      1276 | O             |    194641.93 | 1995-10-23  | 4-NOT SPECIFIED | Clerk#000000259 |              0 | zzle. carefully enticing deposits nag furio
         37 |       862 | F             |    131896.49 | 1992-06-03  | 3-MEDIUM        | Clerk#000000456 |              0 | kly regular pinto beans. carefully unusual waters cajole never
         39 |       818 | O             |    326565.37 | 1996-09-20  | 3-MEDIUM        | Clerk#000000659 |              0 | ole express, ironic requests: ir
         67 |       568 | O             |    182481.16 | 1996-12-19  | 4-NOT SPECIFIED | Clerk#000000547 |              0 | symptotes haggle slyly around the furiously iron
         68 |       286 | O             |    301968.79 | 1998-04-18  | 3-MEDIUM        | Clerk#000000440 |              0 |  pinto beans sleep carefully. blithely ironic deposits haggle furiously acro
         69 |       845 | F             |    204110.73 | 1994-06-04  | 4-NOT SPECIFIED | Clerk#000000330 |              0 |  depths atop the slyly thin deposits detect among the furiously silent accou
         71 |        34 | O             |    260603.38 | 1998-01-24  | 4-NOT SPECIFIED | Clerk#000000271 |              0 |  express deposits along the blithely regul
        100 |      1471 | O             |    198978.27 | 1998-02-28  | 4-NOT SPECIFIED | Clerk#000000577 |              0 | heodolites detect slyly alongside of the ent
        102 |         8 | O             |    184806.58 | 1997-05-09  | 2-HIGH          | Clerk#000000596 |              0 |  slyly according to the asymptotes. carefully final packages integrate furious
        103 |       292 | O             |    118745.16 | 1996-06-20  | 4-NOT SPECIFIED | Clerk#000000090 |              0 | ges. carefully unusual instructions haggle quickly regular f
        129 |       712 | F             |    254281.41 | 1992-11-19  | 5-LOW           | Clerk#000000859 |              0 | ing tithes. carefully pending deposits boost about the silently express
        130 |       370 | F             |    140213.54 | 1992-05-08  | 2-HIGH          | Clerk#000000036 |              0 | le slyly unusual, regular packages? express deposits det
        134 |        62 | F             |    208201.46 | 1992-05-01  | 4-NOT SPECIFIED | Clerk#000000711 |              0 | lar theodolites boos
        135 |       605 | O             |    230472.84 | 1995-10-21  | 4-NOT SPECIFIED | Clerk#000000804 |              0 | l platelets use according t
        163 |       878 | O             |    177809.13 | 1997-09-05  | 3-MEDIUM        | Clerk#000000379 |              0 | y final packages. final foxes since the quickly even
        164 |         8 | F             |    250417.20 | 1992-10-21  | 5-LOW           | Clerk#000000209 |              0 | cajole ironic courts. slyly final ideas are slyly. blithely final Tiresias sub
```

2.9 查询供应了所有类型商品的供应商信息

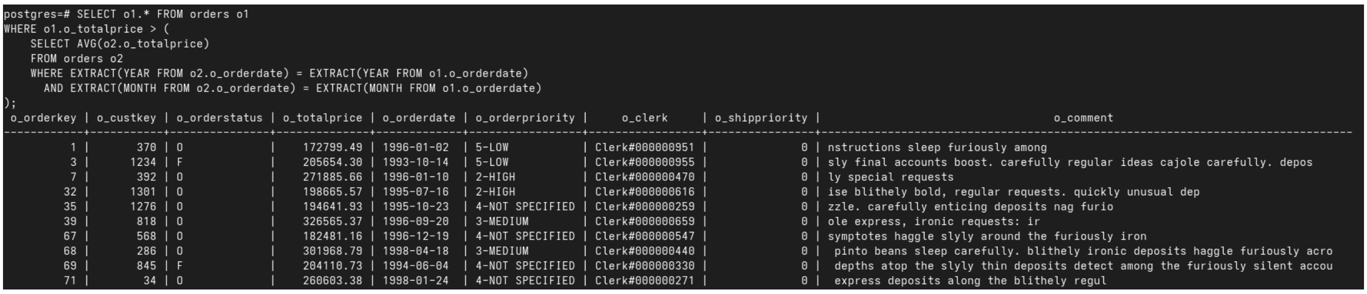
```sql
SELECT s.* FROM supplier s
WHERE NOT EXISTS (
    SELECT p.p_type
    FROM part p
    WHERE NOT EXISTS (
        SELECT 1
        FROM partsupp ps
        JOIN part p2 ON ps.ps_partkey = p2.p_partkey
        WHERE ps.ps_suppkey = s.s_suppkey
          AND p2.p_type = p.p_type
    )
);
```



```
postgres=# SELECT s.* FROM supplier s
WHERE NOT EXISTS (
    SELECT p.p_type
    FROM part p
    WHERE NOT EXISTS (
        SELECT 1
        FROM partsupp ps
        JOIN part p2 ON ps.ps_partkey = p2.p_partkey
        WHERE ps.ps_suppkey = s.s_suppkey
          AND p2.p_type = p.p_type
    )
);
 s_suppkey | s_name | s_address | s_nationkey | s_phone | s_acctbal | s_comment
-----------+--------+-----------+-------------+---------+-----------+-----------
(0 rows)

postgres=#
```

2.10 查询总价大于所处月份平均总价的订单

```sql
SELECT o1.* FROM orders o1
WHERE o1.o_totalprice > (
    SELECT AVG(o2.o_totalprice)
    FROM orders o2
    WHERE EXTRACT(YEAR FROM o2.o_orderdate) = EXTRACT(YEAR FROM
o1.o_orderdate)
      AND EXTRACT(MONTH FROM o2.o_orderdate) = EXTRACT(MONTH FROM
o1.o_orderdate)
);
```



```
postgres=# SELECT o1.* FROM orders o1
WHERE o1.o_totalprice > (
    SELECT AVG(o2.o_totalprice)
    FROM orders o2
    WHERE EXTRACT(YEAR FROM o2.o_orderdate) = EXTRACT(YEAR FROM o1.o_orderdate)
      AND EXTRACT(MONTH FROM o2.o_orderdate) = EXTRACT(MONTH FROM o1.o_orderdate)
);
 o_orderkey | o_custkey | o_orderstatus | o_totalprice | o_orderdate | o_orderpriority |     o_clerk     | o_shippriority |                                  o_comment
------------+-----------+---------------+--------------+-------------+-----------------+-----------------+----------------+-------------------------------------------------------------------------------
          1 |       370 | O             |    172799.49 | 1996-01-02  | 5-LOW           | Clerk#000000951 |              0 | nstructions sleep furiously among
          3 |      1234 | F             |    205654.30 | 1993-10-14  | 5-LOW           | Clerk#000000955 |              0 | sly final accounts boost. carefully regular ideas cajole carefully. depos
          7 |       392 | O             |    271885.66 | 1996-01-10  | 2-HIGH          | Clerk#000000470 |              0 | ly special requests
         32 |      1301 | O             |    198665.57 | 1995-07-16  | 2-HIGH          | Clerk#000000616 |              0 | ise blithely bold, regular requests. quickly unusual dep
         35 |      1276 | O             |    194641.93 | 1995-10-23  | 4-NOT SPECIFIED | Clerk#000000259 |              0 | zzle. carefully enticing deposits nag furio
         39 |       818 | O             |    326565.37 | 1996-09-20  | 3-MEDIUM        | Clerk#000000659 |              0 | ole express, ironic requests: ir
         67 |       568 | O             |    182481.16 | 1996-12-19  | 4-NOT SPECIFIED | Clerk#000000547 |              0 | symptotes haggle slyly around the furiously iron
         68 |       286 | O             |    301968.79 | 1998-04-18  | 3-MEDIUM        | Clerk#000000440 |              0 |  pinto beans sleep carefully. blithely ironic deposits haggle furiously acro
         69 |       845 | F             |    204110.73 | 1994-06-04  | 4-NOT SPECIFIED | Clerk#000000330 |              0 |  depths atop the slyly thin deposits detect among the furiously silent accou
         71 |        34 | O             |    260603.38 | 1998-01-24  | 4-NOT SPECIFIED | Clerk#000000271 |              0 |  express deposits along the blithely regul
```