

数据库系统课程实验报告

|  |  |
| --- | --- |
| 实验名称： | 数据高级查询 |
| 实验日期： | 2023.04.21 |
| 实验地点： | 文宣楼B311 |
| 提交日期： |  |
|  | |
| 学号： | 32420212202930 |
| 姓名： | 陈澄 |
| 专业年级： | 软工2021级 |
| 学年学期： | 2022-2023学年第二学期 |

1. 实验目的

 熟练掌握设计正确的 SQL 查询语句以实现数据高级查询的方法

 熟练掌握 openGauss 连接查询、子查询和集合查询的语法结构及使用方法

-（内）连接、（全）外连接、左外连接、右外连接

-子查询（嵌套查询）

-不相关子查询与相关子查询

-EXISTS/NOT EXISTS

-ANY

-ALL

-集合运算：UNION、INSERSECT、MINUS/EXCEPT

 理解不相关子查询与相关子查询的不同，掌握构造相应 SQL 语句的方法

 熟练掌握基于派生表的查询方法

1. 实验内容和步骤

（1）openGuass 的连接查询语法：

创建两张表 palette\_a 和 palette\_b（结构相同，但表名不同，color 为颜色）

CREATE TABLE palette\_a

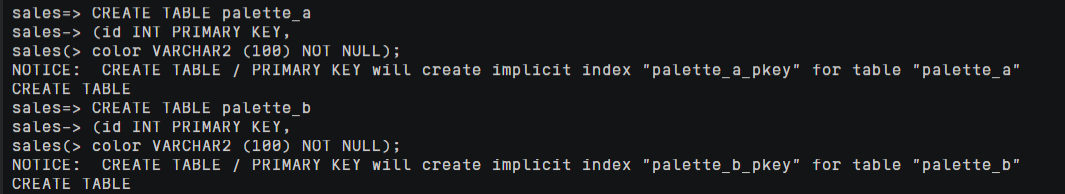
(id INT PRIMARY KEY,

color VARCHAR2 (100) NOT NULL);

CREATE TABLE palette\_b

(id INT PRIMARY KEY,

color VARCHAR2 (100) NOT NULL);



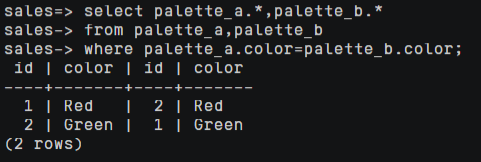
1. 为表 palette\_a 添加样例数据：{(1, 'Red'), (2, 'Green'), (3, 'Blue'), (4, 'Purple')}。



1. 为表 palette\_b 添加样例数据：{(1, 'Green'), (2, 'Red'), (3, 'Cyan'), (4, 'Brown')}。



1. 查询两张表中相同颜色的所有信息。

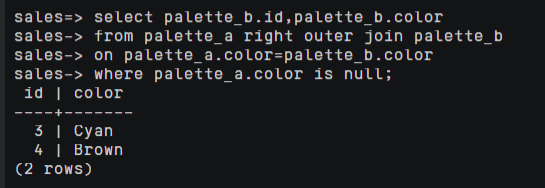


1. 查询 palette\_a 表中颜色不出现在 palette\_b 表中的 id 和颜色。



实现要求：左外连接（必须）+其它查询方法（如果找到）

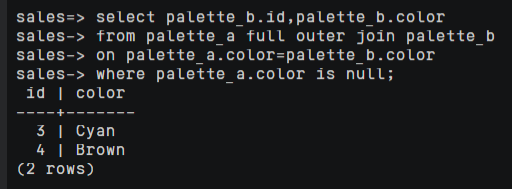
1. 查询 palette\_b 表中颜色不出现在 palette\_a 表中的 id 和颜色。



实现要求：右外连接（必须）+其它查询方法（如果找到）

1. 查询（5）或（6）两种情况的信息（用（全）外连接）。

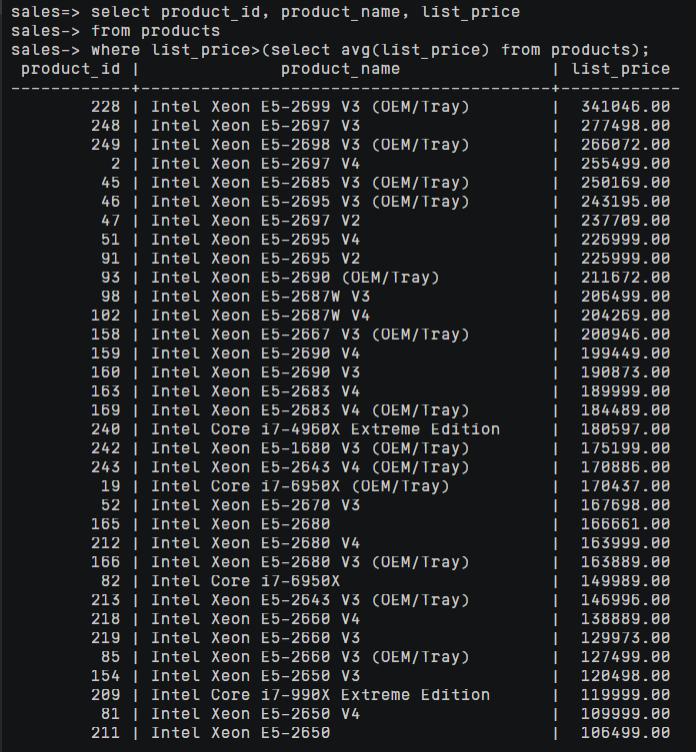


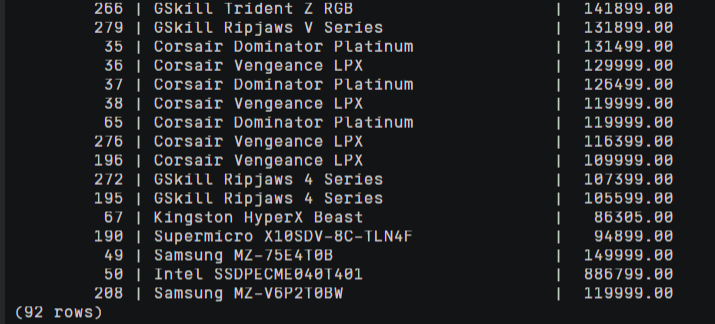


 子查询（Subquery）

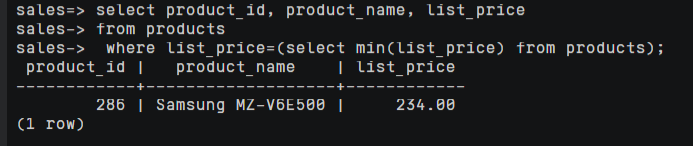
（8）查询产品表 products 中的 product\_id, product\_name, list\_price 信息，要求产品定价

list\_price 大于其平均定价 list\_price。

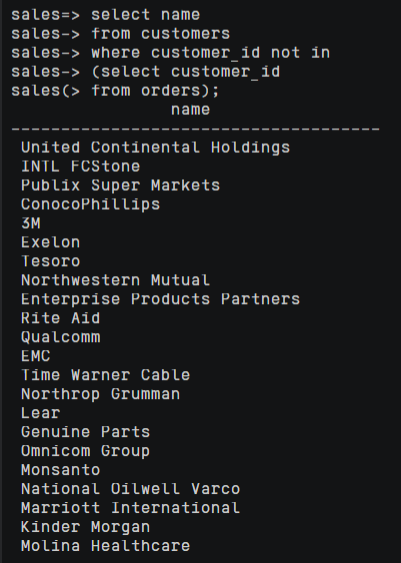


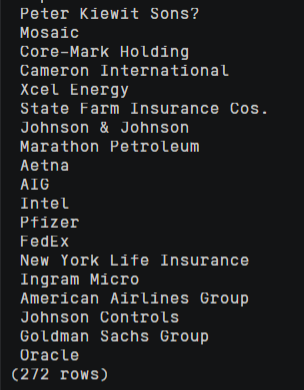


（9）查询产品表 products 中最便宜产品的 product\_id, product\_name, list\_price。



1. 查询没有一个订单的顾客姓名。



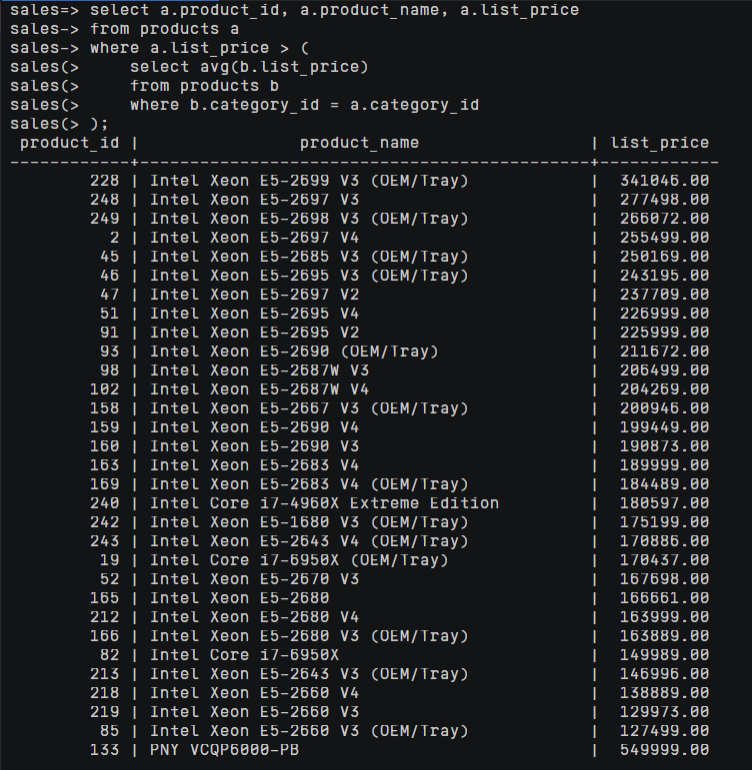


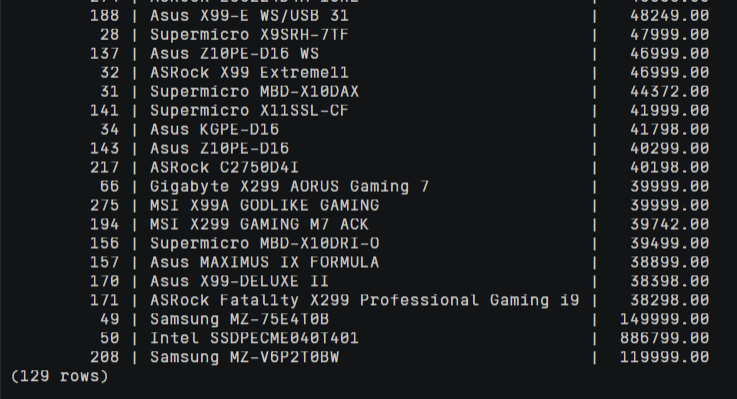
实现要求：NOT IN（必须）+其它查询方法（如果找到）

 相关子查询（correlated subquery）

（11） 查询产品表 products 中产品的 product\_id, product\_name, list\_price，要求产品定价

list\_price 大于其同类产品（可由 category\_id 表达）的平均定价。

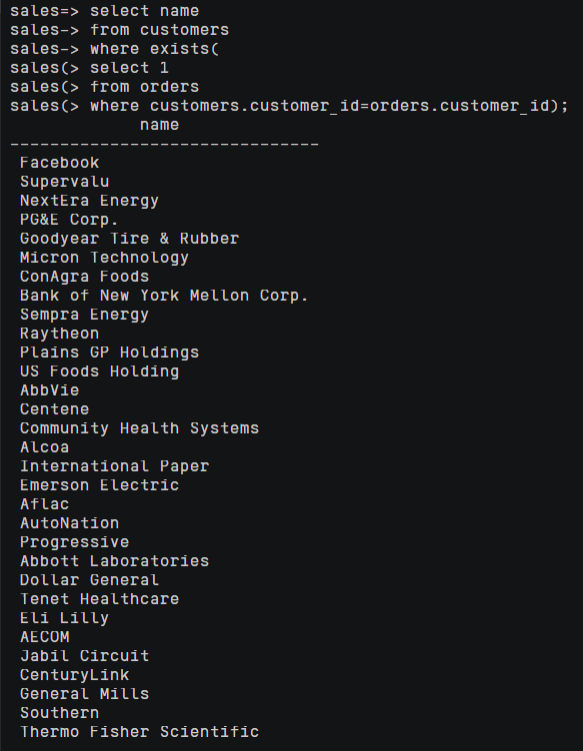


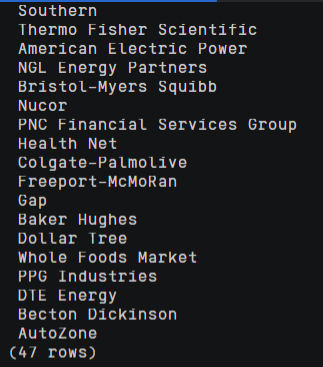


实现要求：相关子查询（必须）+基于派生表的查询（如果找到）

 EXISTS 的使用

（12）查询有订单 order 的所有顾客 customer 姓名（查询涉及 customers 表和 orders 表）。



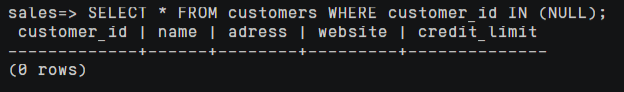


实现要求：使用 EXISTS（必须）+其它查询方法（如果找到）

 EXISTS 与 IN 的不同

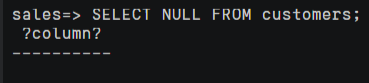
1. 执行以下三条与 NULL 相关的语句，观察各自执行的结果，能否从中得出某些初步结论？

SELECT \* FROM customers WHERE customer\_id IN (NULL);



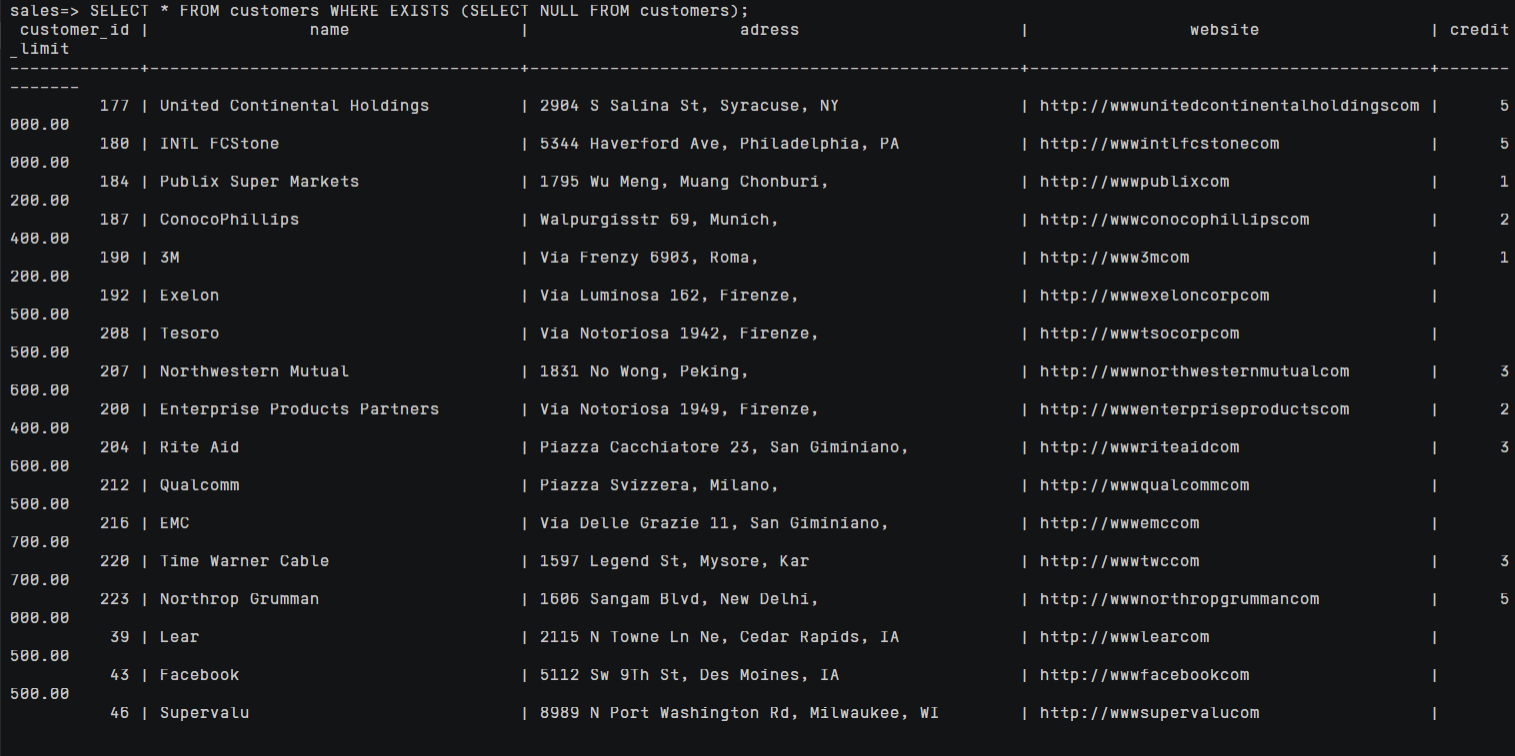
该语句查询了名为 customers 的表中，所有 customer\_id 列的值为 NULL 的记录。但是这样的查询没有实际意义，因为 IN 操作符需要在括号中提供一个包含多个值的列表，而这里只提供了一个 NULL 值，并且不可能与任何其他值匹配。所以该查询将返回一个空结果集。

SELECT NULL FROM customers;



该语句查询了名为 customers 的表中的所有记录，并只检索了一个 NULL 值列。由于在每个记录中都不存在任何列或值与 NULL 列相匹配，因此该查询将返回一列 NULL 值。

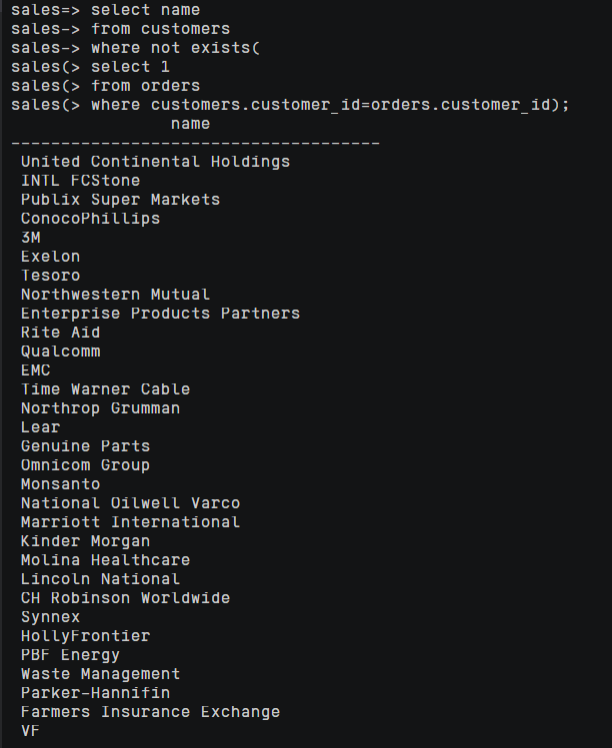
SELECT \* FROM customers WHERE EXISTS (SELECT NULL FROM customers);

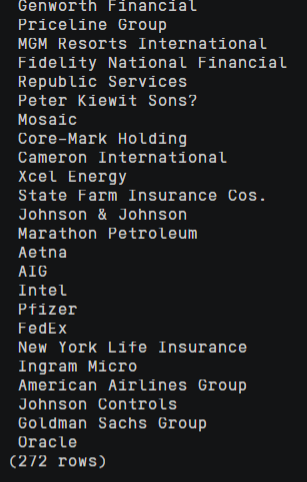


该语句查询了名为 customers 的表中，是否存在记录。其中，内部的子查询 SELECT NULL FROM customers 将返回一个空结果集，但是由于使用了 EXISTS 操作符，所以该查询将始终返回所有存在于表中的记录。

 NOT EXISTS 的使用

1. 找出所有没有订单的顾客姓名（查询涉及 customers 表和 orders 表）。



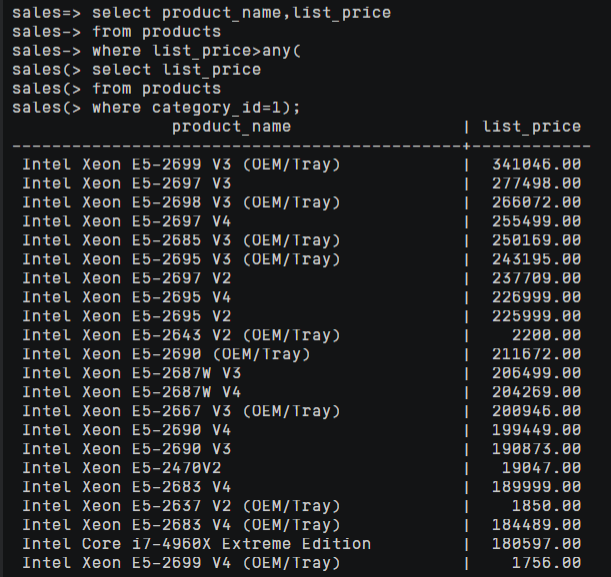


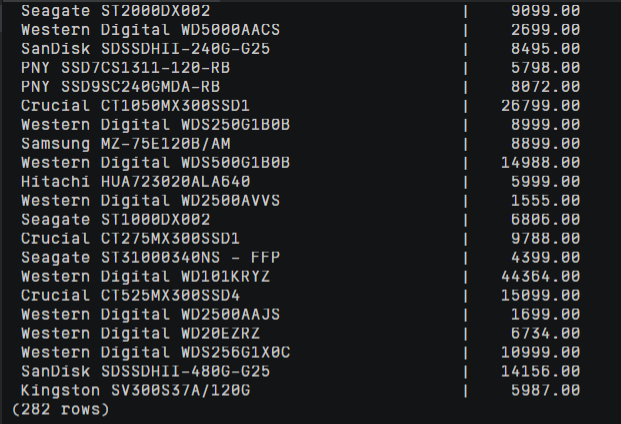
实现要求：使用 NOT EXISTS（必须）+其它查询方法（如果找到）

 ANY 的使用

（15）查询产品表 products 中的产品名 product\_name 和定价 list\_price，要求其定价高于产品种

类 1 中的任何产品定价。



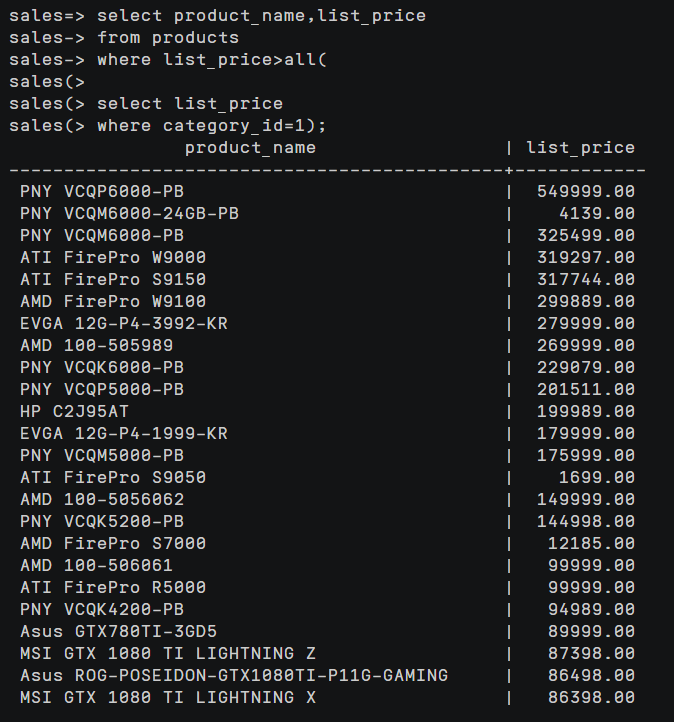


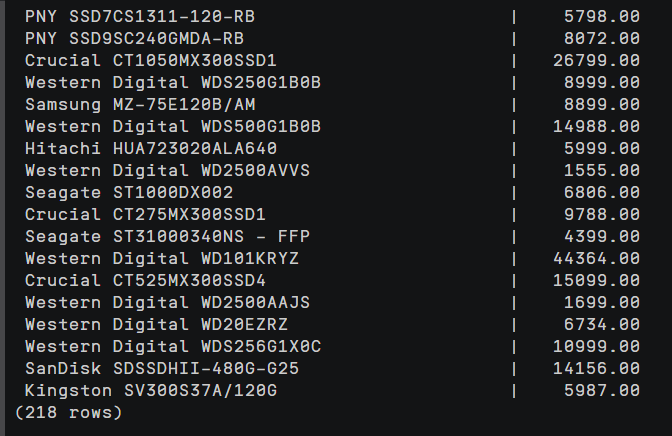
实现要求：ANY（必须）+其它查询方法（如果找到）

 ALL 的使用

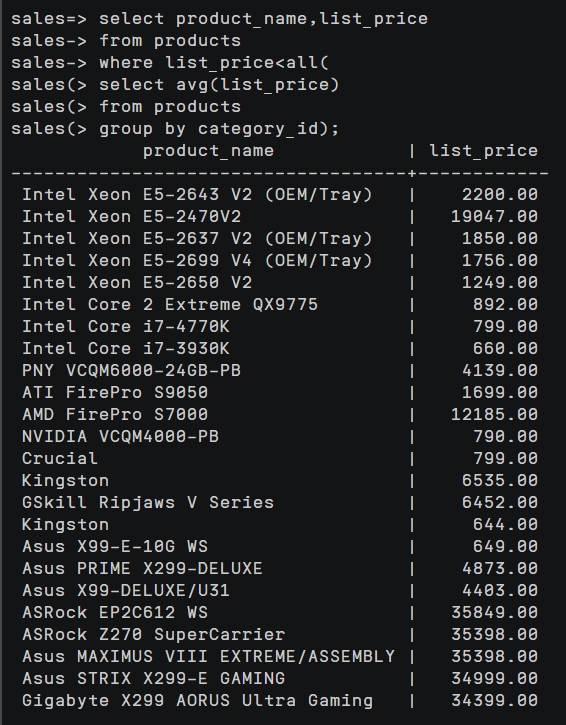
（16）查询产品表 products 中的产品名 product\_name 和定价 list\_price，要求其定价高于产品种

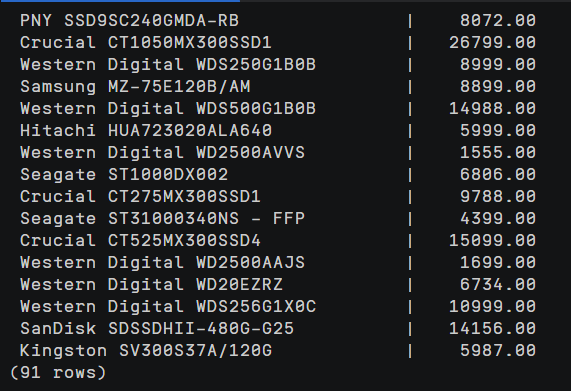
类 1 中的所有定价。





1. 查询产品表 products 中的产品名 product\_name 和定价 list\_price，要求其定价低于产品种类的所有平均定价。





实现要求：ALL（必须）+其它查询方法（如果找到）

 UNION 的使用

1. 查询 contacts 表和 employees 表中的所有 last\_name，并以 last\_name 升序显示。





实现要求：去重+UNION（必须）+其它查询方法（如果找到）

1. 查询 contacts 表和 employees 表中的所有 last\_name，并以 last\_name 升序显示。





实现要求：保留重复+UNION ALL（必须）+其它查询方法（如果找到）

 INTERSECT 的使用

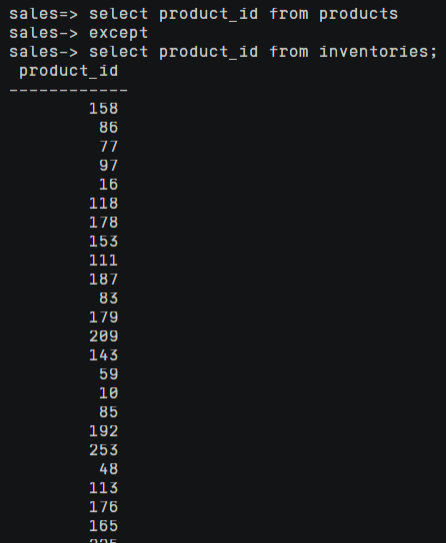
1. 查询同时出现在 contacts 表和 employees 表中的所有 last\_name。

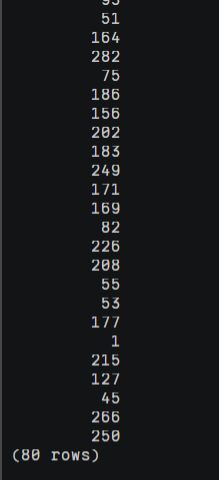


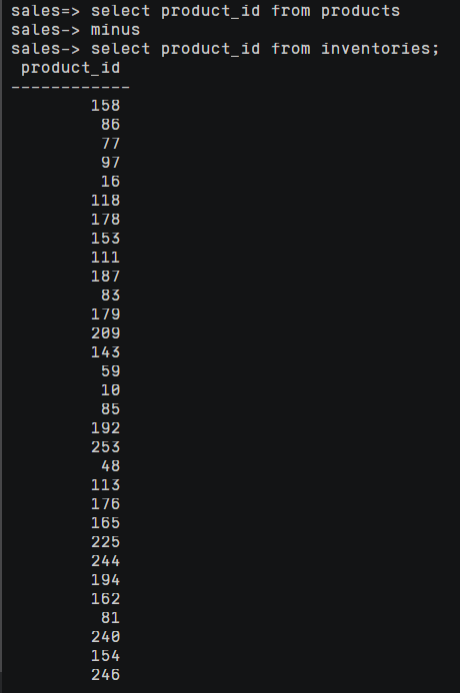
实现要求：INTERSECT（必须）+其它查询方法（如果找到）

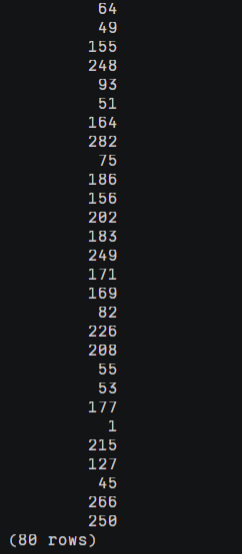
 MINUS/EXCEPT 的使用

1. 查询在产品表 products 中而不在库存表 inventories 中的产品号 product\_id。









实现要求：MINUS/EXCEPT（必须）+其它查询方法（如果找到）

1. 实验总结

3.1 完成的工作

实现了表的连接查询，嵌套查询，集合查询等各种查询方法。

使用了exists,any,all等关键词对各表进行查询。

3.2 对实验的认识

1.学会了表的连接查询，以及内外连接，左右连接的区别与联系等等。

2.学会了表的嵌套查询，相关子查询和不相关子查询的应用以及对其应用场景的判断等。

3.学会了运用exists,not exists两个关键词来实现数据的过滤。

4.学会了运用any,all来比较一个值与子查询中的多个返回值之间的关系。

5.学会了表的集合查询，包括UNION、INTERSECT以及MINUS/EXCEPT。

3.3 遇到的困难及解决方法

无