openGauss 安全体系创新

实践课



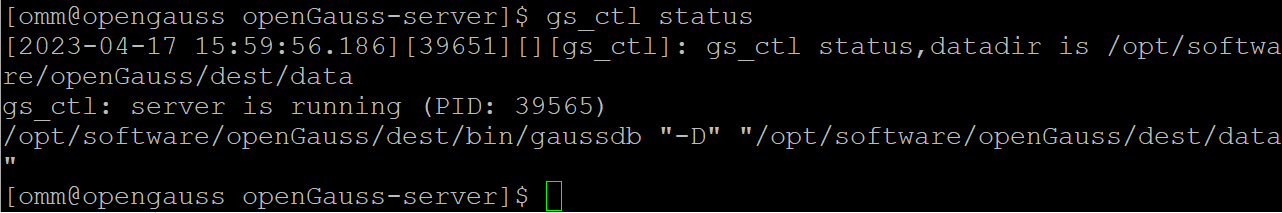
华为技术有限公司

# 关卡一、openGauss数据安装及基本操作

openGauss数据安装及基本操作, 作业提交任务如下：

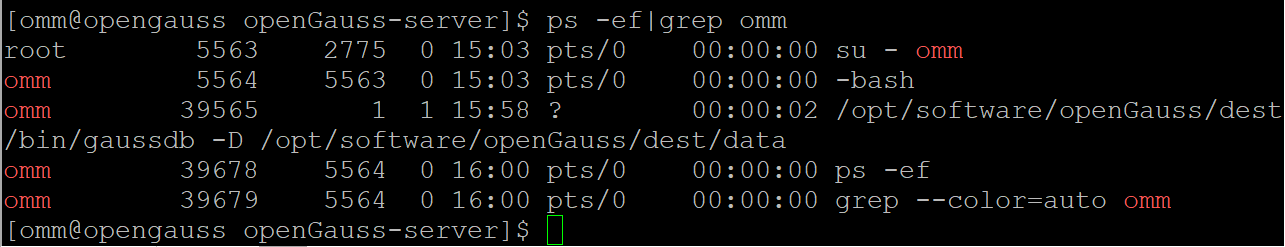
任务一：数据库状态验证

1. 查询数据库状态成功截图



任务二：数据库服务进程验证

1. 查看数据库服务进程截图（包含数据库服务器的主机名）



任务三：实践思考题

思考题1：为什么需要通过源码编译，安装数据库？

答：

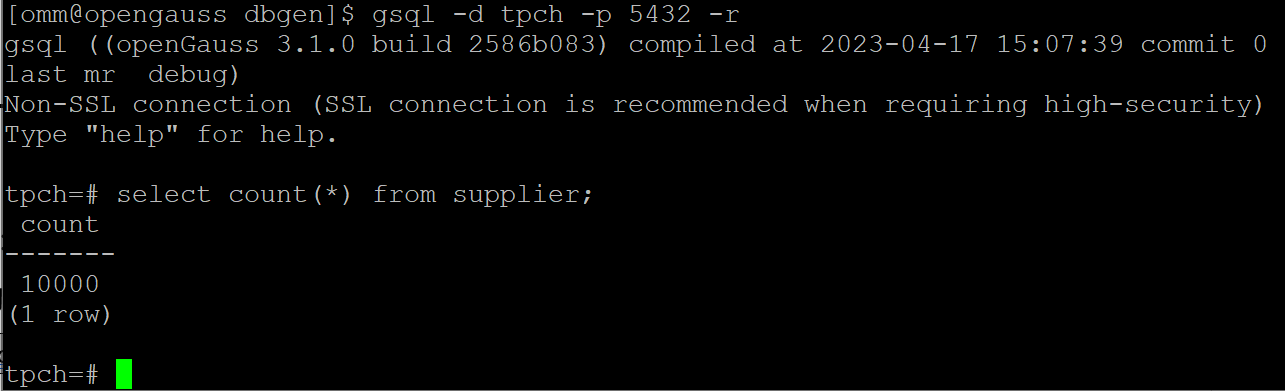
源码编译和安装数据库是为了能够自定义数据库的配置和功能，以及确保数据库的安全性和稳定性。可以实现以下的几个功能：①定制化配置：编译和安装数据库可以允许用户选择所需的功能和选项，比如启用或禁用特定的模块或插件，配置数据库的缓存大小或性能优化等。②安全性和稳定性：源码编译可以确保数据库的安全性和稳定性，因为它可以自动应用最新的安全补丁和修复程序，确保数据库的可靠性和免受潜在的漏洞或攻击。③性能优化：源码编译可以让用户对数据库的性能进行更细粒度的控制，比如编译特定的CPU指令集，配置数据库的缓存和缓存大小等，从而提高数据库的性能和响应速度。④跨平台支持：源码编译可以确保数据库在不同的平台和操作系统上运行良好，因为它可以根据不同的操作系统和硬件架构进行编译，从而最大限度地提高兼容性。总之，源码编译和安装数据库可以让用户获得更大的自由度和控制权，同时也可以提高数据库的性能、安全性和稳定性。

# 关卡二、openGauss数据导入及行存列存

任务一：数据初始化验证

1. 查询supplier表的行数，并将结果进行图：

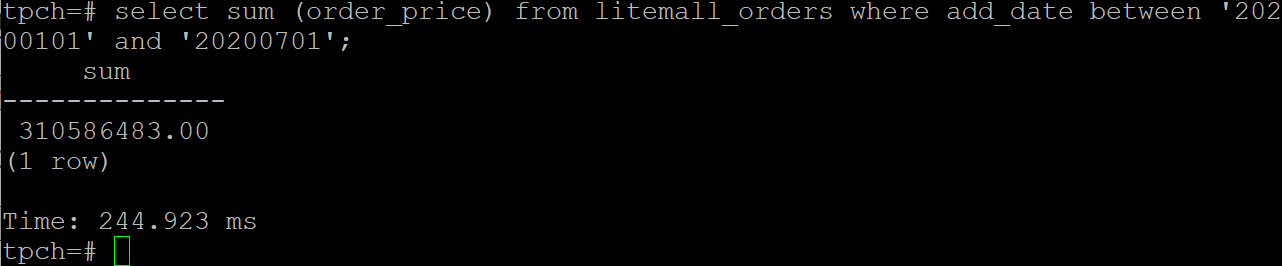
select count(\*) from supplier;;



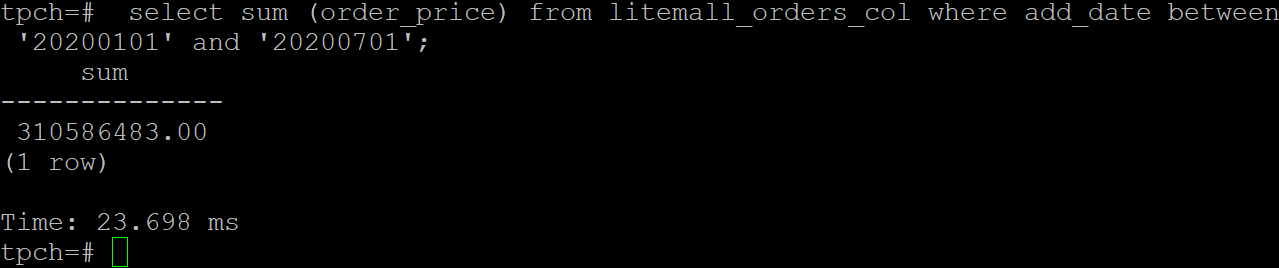
任务二：行存表与列存表执行效率对比

1. 2020年上半年litemall\_orders行存表与litemall\_orders\_col列存表中的order\_price的总和查询，并对比执行效率截图

select sum (order\_price) from litemall\_orders where add\_date between '20200101' and '20200701';

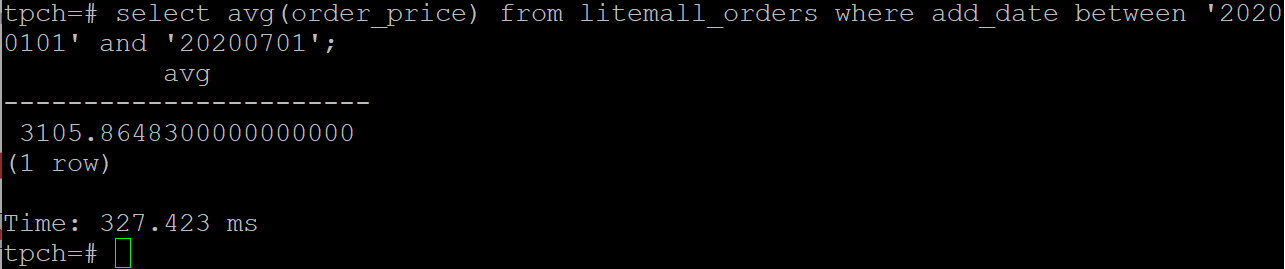


select sum (order\_price) from litemall\_orders\_col where add\_date between '20200101' and '20200701';

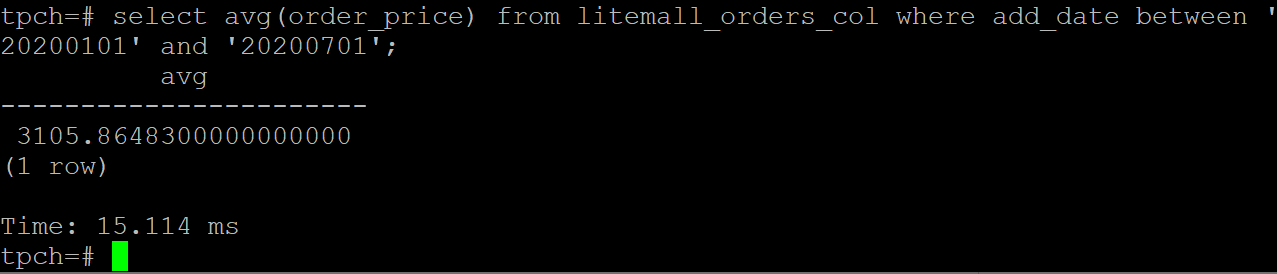


2. 2020年上半年litemall\_orders行存表与litemall\_orders\_col列存表中的order\_price的平均值查询，并对比执行效率截图

select avg (order\_price) from litemall\_orders where add\_date between '20200101' and '20200701';

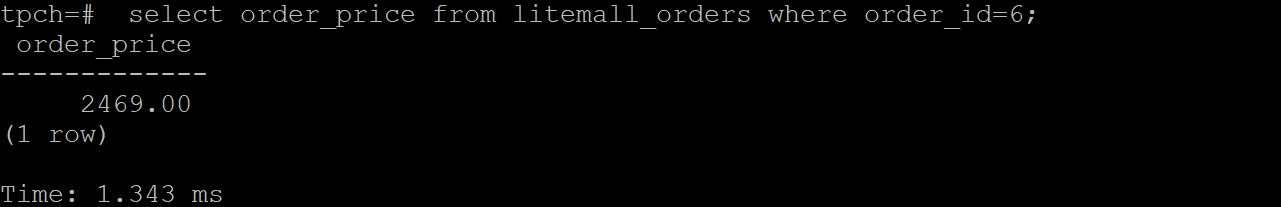


select avg (order\_price) from litemall\_orders\_col where add\_date between '20200101' and '20200701';

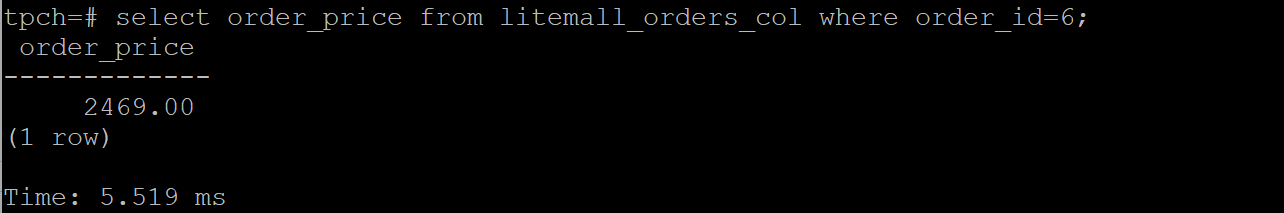


3. 查询litemall\_orders行存表与litemall\_orders\_col列存表中order\_id为6的order\_price的值，并对比执行效率截图。

select order\_price from litemall\_orders where order\_id=6;

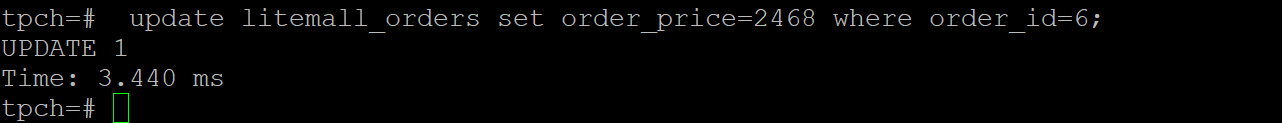


select order\_price from litemall\_orders\_col where order\_id=6;

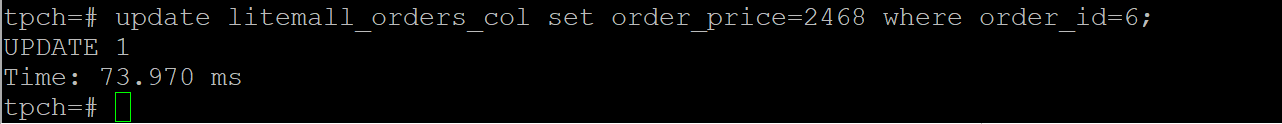


4. 将litemall\_orders行存表与litemall\_orders\_col列存表中order\_id为6的order\_price修改为2468，并对比执行效率截图。

update litemall\_orders set order\_price=2468 where order\_id=6;



update litemall\_orders\_col set order\_price=2468 where order\_id=6;



任务三：实践思考题

思考题1：

行存表与列存表在执行相同的SQL语句时，为何执行的时间不同？

答：在 OpenGauss 数据库中，行存表和列存表是两种不同的存储引擎，它们的数据存储方式和读取方式不同，所以在执行相同的 SQL 语句时，执行时间也会有所不同。行存表是将整个行数据存储在一起，这种方式适合对数据的增删改操作较多的场景，因为每次增删改都只需要对一行数据进行操作。但是在查询操作中，由于需要扫描整个行数据，所以查询效率相对较低。列存表是将同一列的数据存储在一起，这种方式适合对数据的查询操作较多的场景，因为在查询时只需要扫描需要的列数据，所以查询效率相对较高。但是在增删改操作中，由于需要更新多个列数据，所以效率相对较低。在执行相同的 SQL 语句时，如果是针对查询操作，则列存表的执行时间往往比行存表的执行时间更快，因为列存表的数据组织方式可以最大限度地减少数据的读取量。而如果是针对增删改操作，则行存表的执行时间往往比列存表的执行时间更快，因为行存表的数据组织方式可以最大限度地减少数据的写入量。需要根据具体是业务场景选择合适的操作方式。

思考题2：

在执行哪些类型SQL时，行存表效率更高？在执行哪些类型SQL时，列存表效率更高？

答：行存表效率高的情况：

增删改操作：由于行存表是将整个行数据存储在一起，所以在执行增删改操作时，只需要修改一行数据，相对来说更新的数据量更小，因此行存表效率更高。

小型事务操作：在小型事务中，涉及的数据量较少，因此行存表比列存表更加适用。

随机IO操作：行存表的数据组织方式比较紧凑，随机 IO 操作的次数较少，因此行存表在处理随机 IO 时的效率更高。

列存表的效率高的情况：

聚合查询：由于列存表是将同一列的数据存储在一起，因此在执行聚合查询时，只需要扫描需要的列数据，可以减少数据的读取量，因此列存表效率更高。

大规模数据查询：在查询大规模数据时，列存表的数据组织方式可以最大程度地减少数据的读取量，因此列存表在处理大规模数据查询时效率更高。

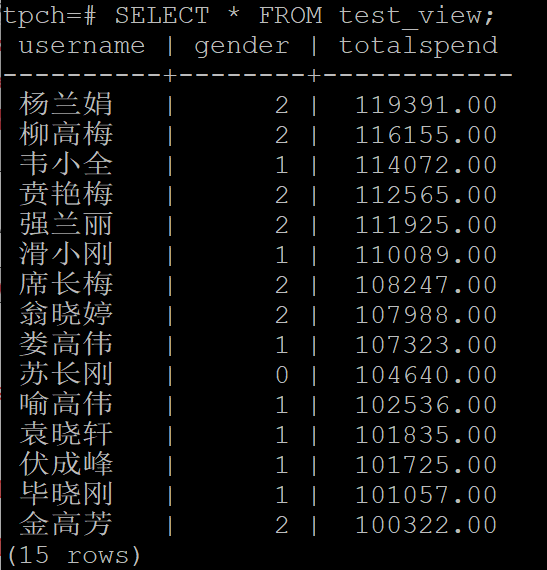
读密集型操作：在读密集型操作中，查询操作比较频繁，而列存表的数据组织方式可以最大程度地减少数据的读取量，因此列存表在读密集型操作中效率更高。

# 关卡三：openGauss物化视图应用

任务一：物化视图的使用

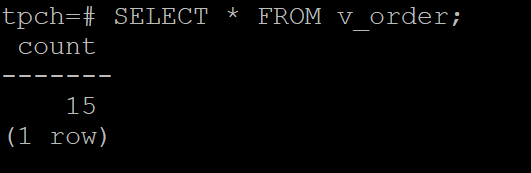
1. 创建物化视图所需要的表后，对表内容进行查询，对查询结果截图：

SELECT \* FROM test\_view;



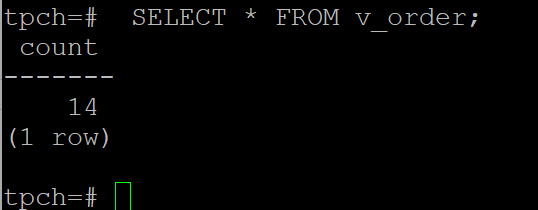
2. 使用物化视图统计人数，查询物化视图结果，将执行结果截图。

SELECT \* FROM v\_order;



3. 对表进行操作后，刷新物化视图，查询物化视图结果，将执行结果截图。

SELECT \* FROM v\_order;



4. 创建增量物化视图，查询物化视图结果，将执行结果截图。

SELECT \* FROM vi\_order;



5. 对表进行操作后，刷新增量物化视图，查询物化视图结果，将执行结果截图。

SELECT \* FROM vi\_order;





任务二：实践思考题

思考题1：全量物化视图与增量物化视图有哪些差别？

答：全量物化视图会在创建时将查询结果全部缓存到物化视图中，这样每次查询时都可以直接从物化视图中获取数据。当源表数据发生变化时，需要手动刷新物化视图，这样才能保证物化视图中的数据和源表中的数据一致。全量物化视图适用于数据变化不频繁的情况，因为每次变化都需要重新刷新整个物化视图，刷新成本较高。增量物化视图会在创建时将查询结果的一部分缓存到物化视图中，当源表数据发生变化时，增量物化视图会只刷新发生变化的部分数据，而不是整个物化视图。因此，增量物化视图适用于数据变化比较频繁的情况，因为只需要刷新部分数据，刷新成本相对较低。如果数据变化不频繁，且需要快速地查询大量数据，可以选择全量物化视图；如果数据变化频繁，需要快速地查询部分数据，可以选择增量物化视图。

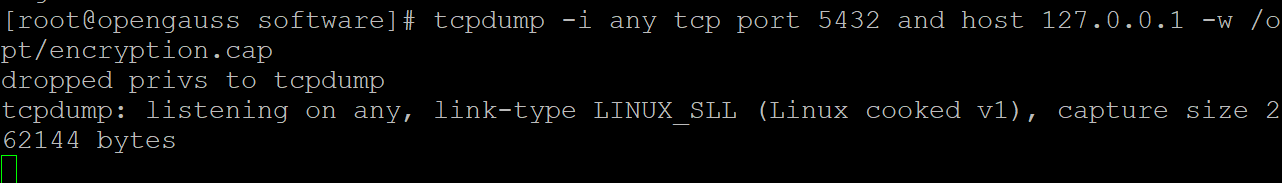
思考题2：物化视图适用那些使用场景？

答：物化视图适用于需要频繁进行复杂查询操作、实时计算和分析、数据统计和报表生成等场景。

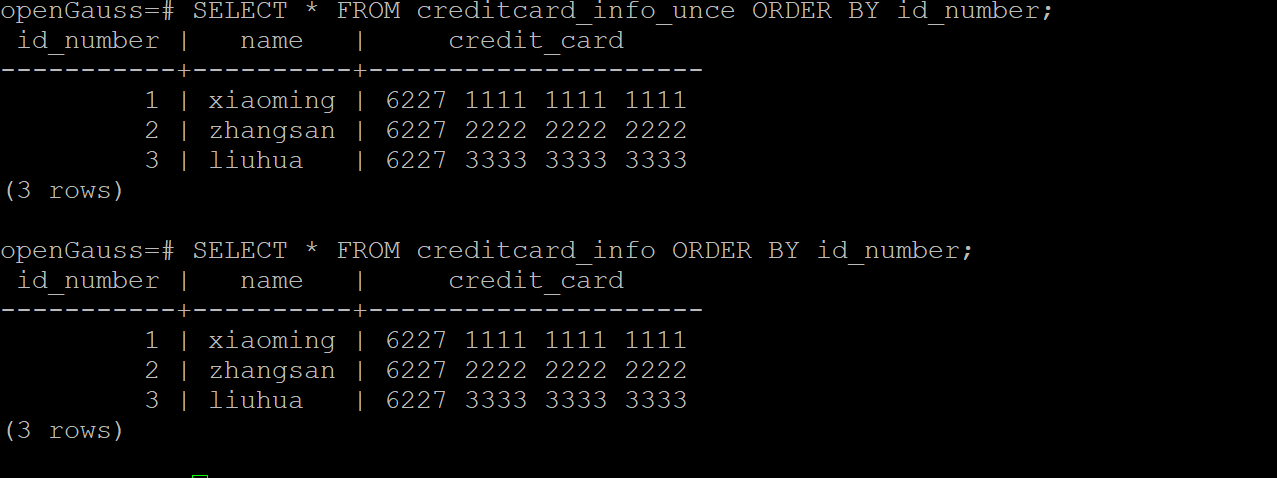
# 关卡四：openGauss密态数据库特性应用

任务一：物化视图的使用

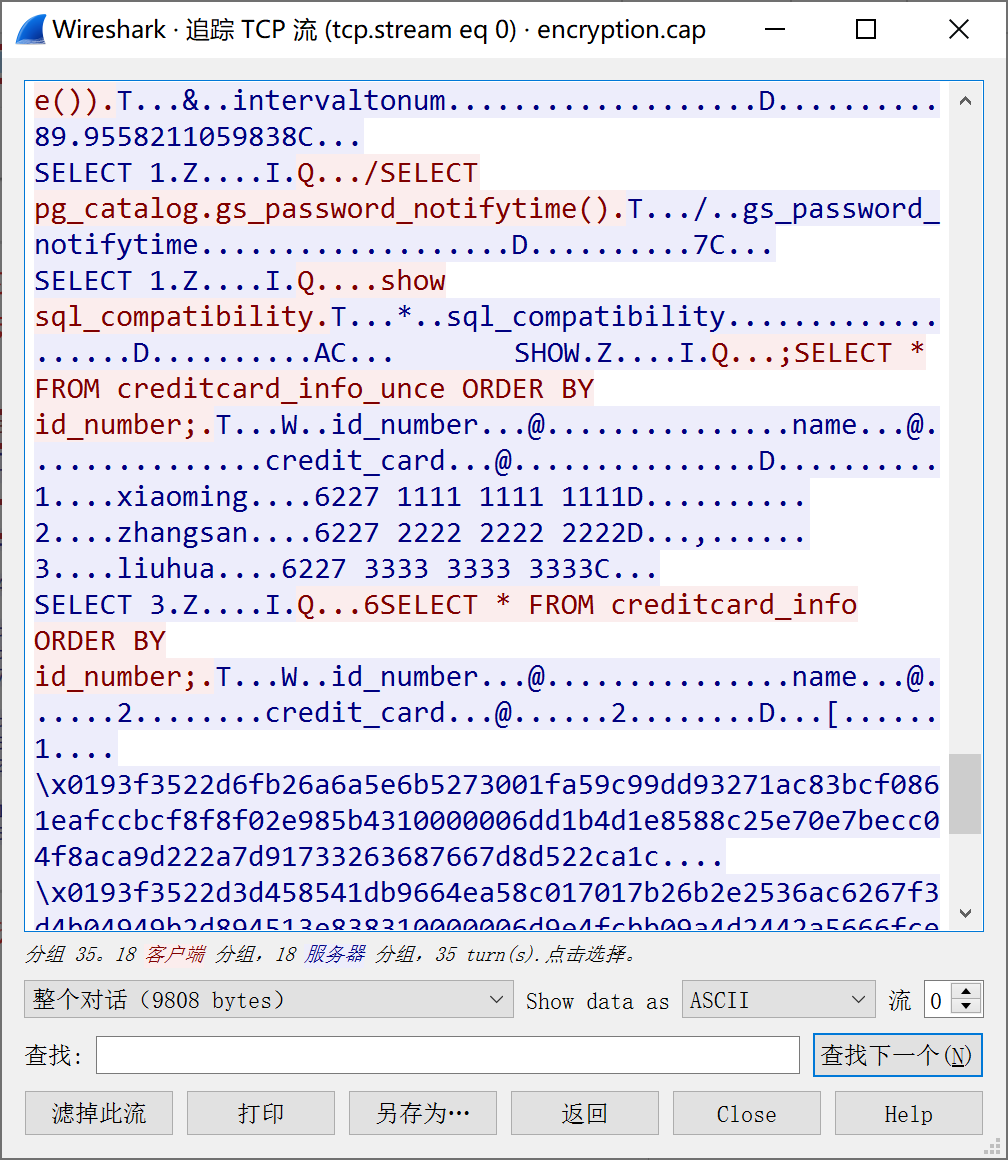
1. 通过tcpdump抓取数据流，此putty窗口暂时保持不动，将执行结果截图：



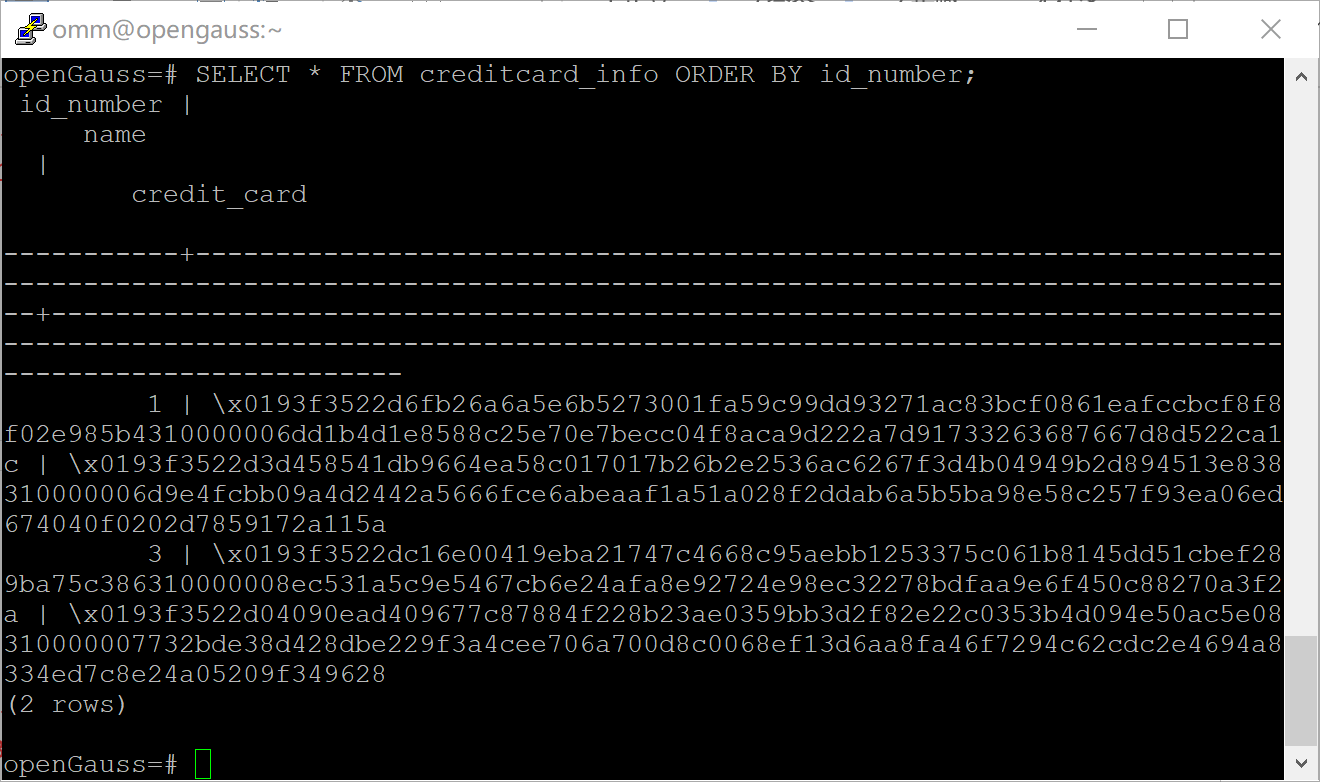
1. 将加密表和非加密表查询结果截图：



1. 用wireshark解析加密表和非加密表的差异时，非加密表name列和credit\_card列是明文，加密表name列和credit\_card列均是密文，将执行结果截图：



1. 查询加密表，查询到的结果为密文，将执行结果截图：



任务二：实践思考题

思考题1：

数据实际存储在物理磁盘上的时候是明文还是密文？数据的加解密的动作是在客户端完成的还是服务端完成的？

答：数据存储在物理磁盘上时是密文形式，当数据被写入数据库时，它会被加密后再存储到磁盘上。因此，对于非授权用户来说，他们无法直接从磁盘上读取到明文数据，而只能读取到密文数据。所有的加密和解密操作都由数据库内部完成，而不需要客户端进行额外的加密和解密操作。这样可以保证数据的安全性和完整性，减少了客户端的负担，同时提高了数据传输的效率和速度。