openGauss 安全体系创新

实践课



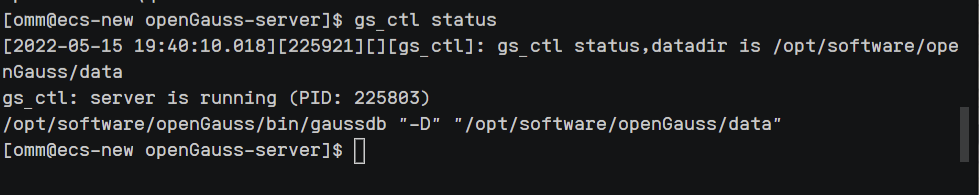
华为技术有限公司

# 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：

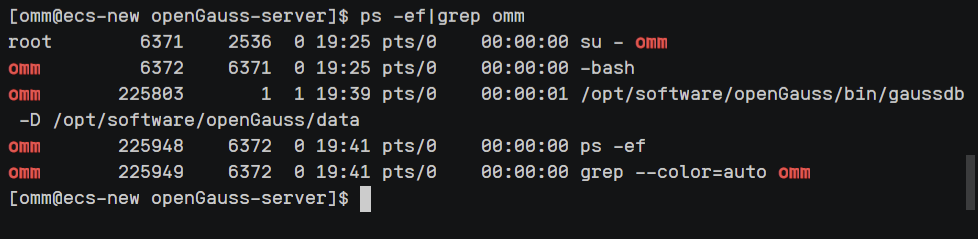
任务一：数据库状态验证

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：因为为了适配不同系统环境等，还有能使用最新最完整版本，使用源码编译完成数据库安装可以避免很多不同系统环境兼容问题。

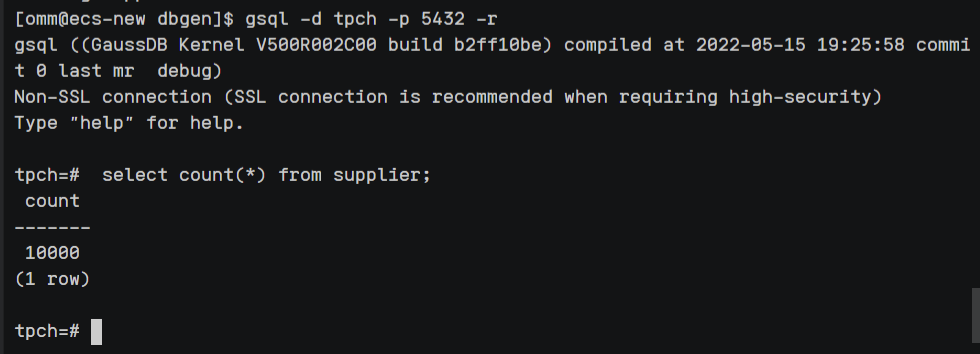
安装步骤：1.创建服务器系统，2.配置系统编译环境，3.下载数据库源码，4.解压配置安装文件，5.安装，6.测试是否安装成功。

# 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：

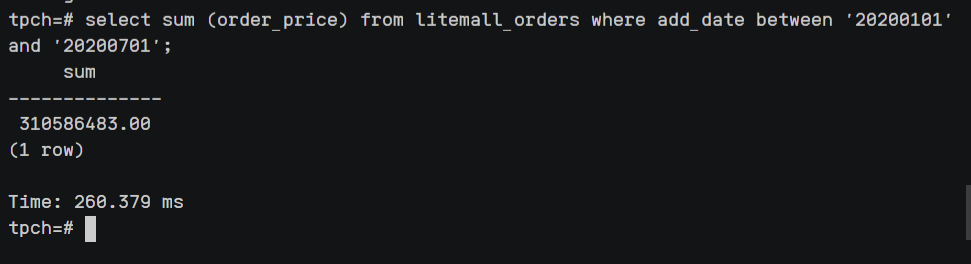
select count(\*) from supplier;;



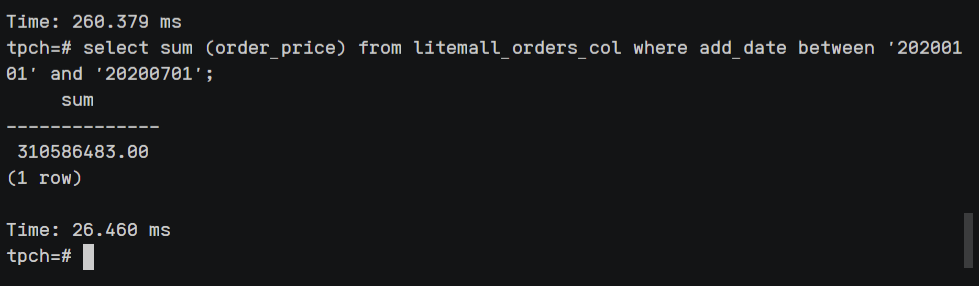
任务二：行存表与列存表执行效率对比

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';



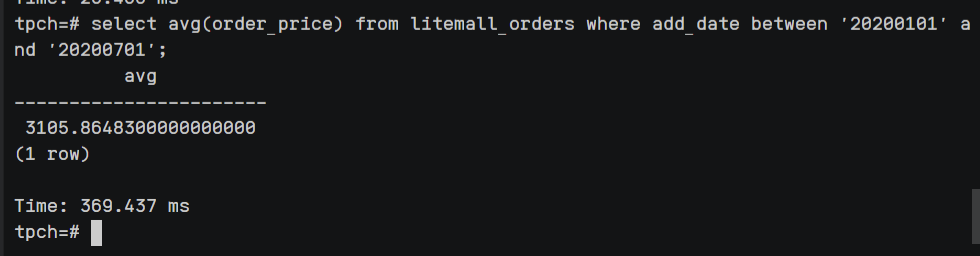
select sum (order\_price) from litemall\_orders\_col where add\_date between '20200101' and '20200701';



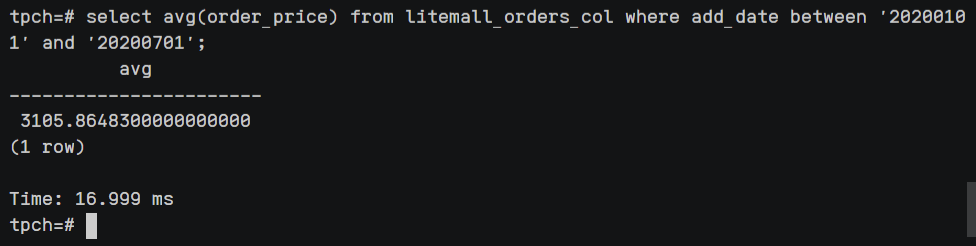
可以看出列存表在求和上比行存表快。

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';



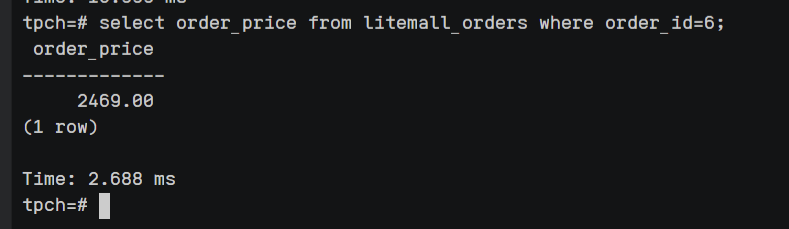
select avg (order\_price) from litemall\_orders\_col where add\_date between '20200101' and '20200701';



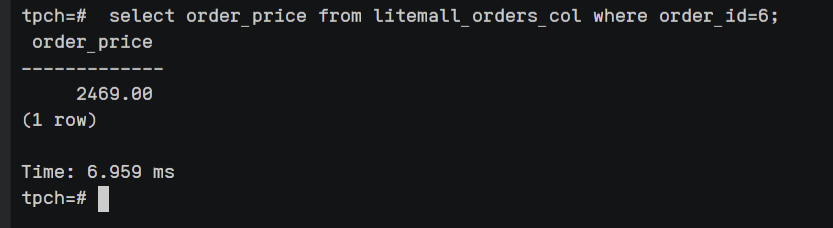
可以看出列存表在求均值上比行存表快。

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;



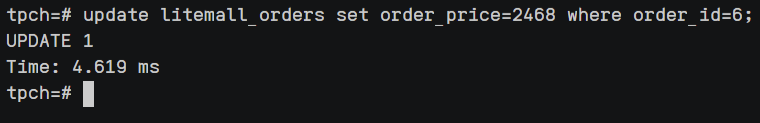
select order\_price from litemall\_orders\_col where order\_id=6;



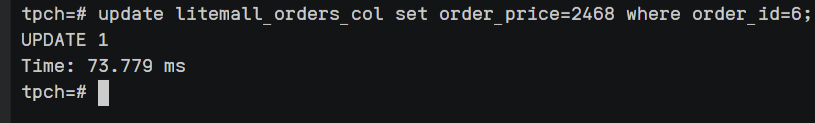
可以看出行存表在查询上比列存表更快。

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;



可以看出行存表在更新数据上比列存表更快。

所以综上可得，在有大量求和求平均等查询时，使用列存表更好；而在有大量取出数据查询，更新数据的情景下，使用行存表更好。

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：行存储的写入是一次完成，数据的完整性因此可以确定。列存储需要把一行记录拆分成单列保存，写入次数明显比行存储多。行存储在写入上占有很大的优势。行存储的写入是一次性完成，消耗的时间比列存储少，并且能够保证数据的完整性，所以在更新数据操作时会更快，缺点是数据读取过程中，由于每次读取完整一行所有属性，所以会产生冗余数据。列存储在写入效率、保证数据完整性上都不如行存储，它的优势是在读取过程，不会产生冗余数据，每次读取同一属性列的所有，这对数据完整性要求不高的大数据处理领域（如求和求均值）更好。

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：在更新写入数据和查询单条数据时，行存表效率更高。在对单一列属性求和、求均值等列属性操作上，列存表效率更高。

# 关卡三：openGauss物化视图应用

任务一：物化视图的使用

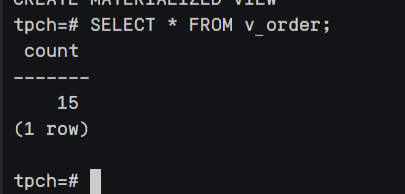
1. 创建物化视图所需要的表后，对表内容进行查询，对查询结果截图：

SELECT \* FROM test\_view;



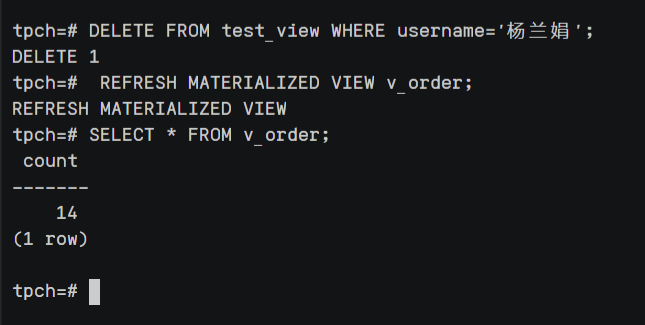
2. 使用物化视图统计人数，查询物化视图结果，将执行结果截图。

SELECT \* FROM v\_order;



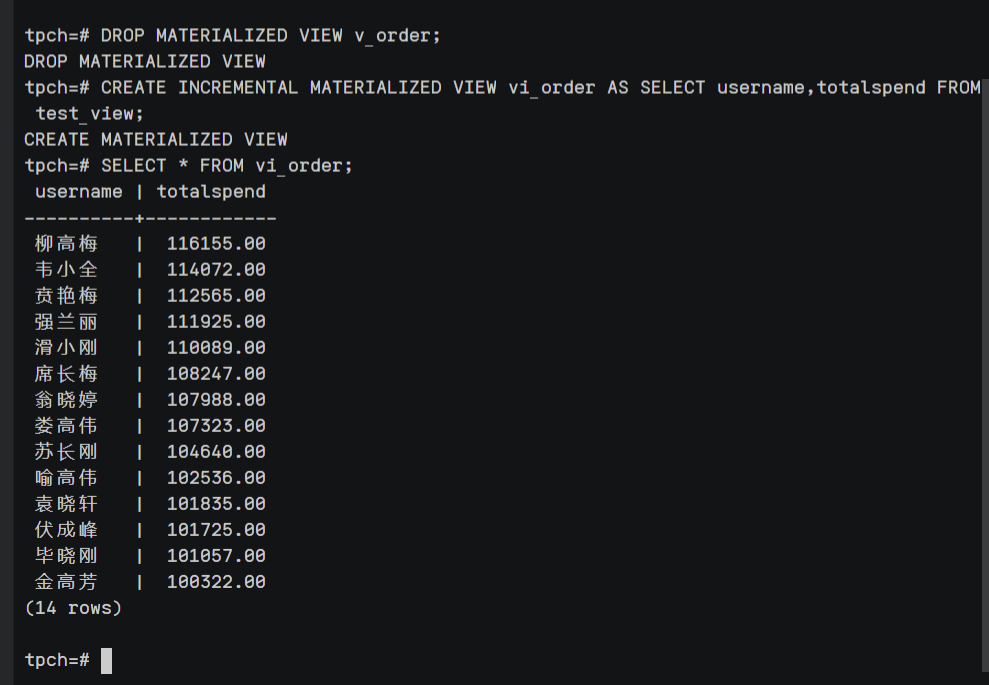
3. 对表进行操作后，刷新物化视图，查询物化视图结果，将执行结果截图。

SELECT \* FROM v\_order;



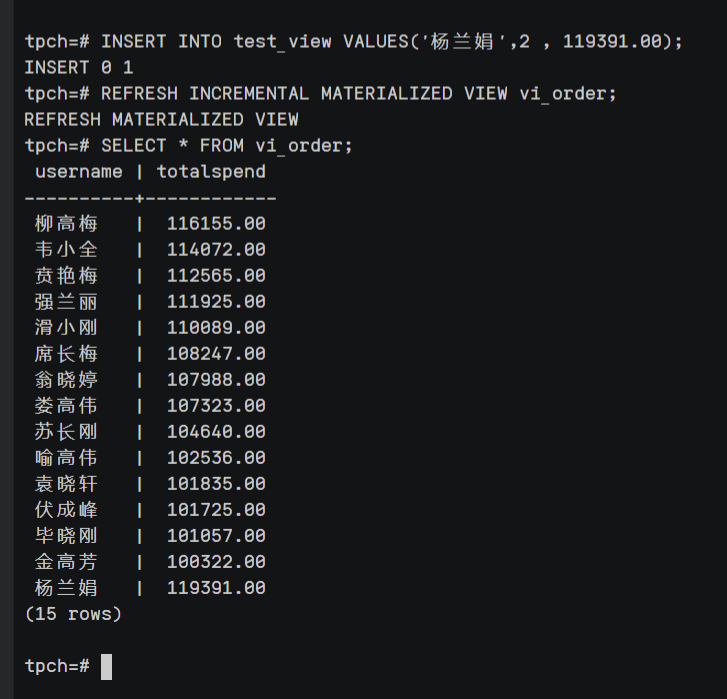
4. 创建增量物化视图，查询物化视图结果，将执行结果截图。

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：全量每次刷新时会全量刷新，是所有数据都会刷新然后同步，可以用来统计总体变化；而增量是每次有变动了后，仅刷新变动的地方，适合大量更新刷新。

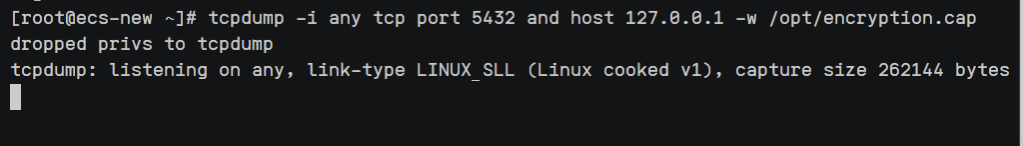
思考题2：物化视图适用那些使用场景？

答：1. 用于查询优化，可以更方便的查询数据，保证数据一致性。2.用于高级的修改，复制；不用对整个表或数据库操作，仅需选取一个物化视图进行操作即可。

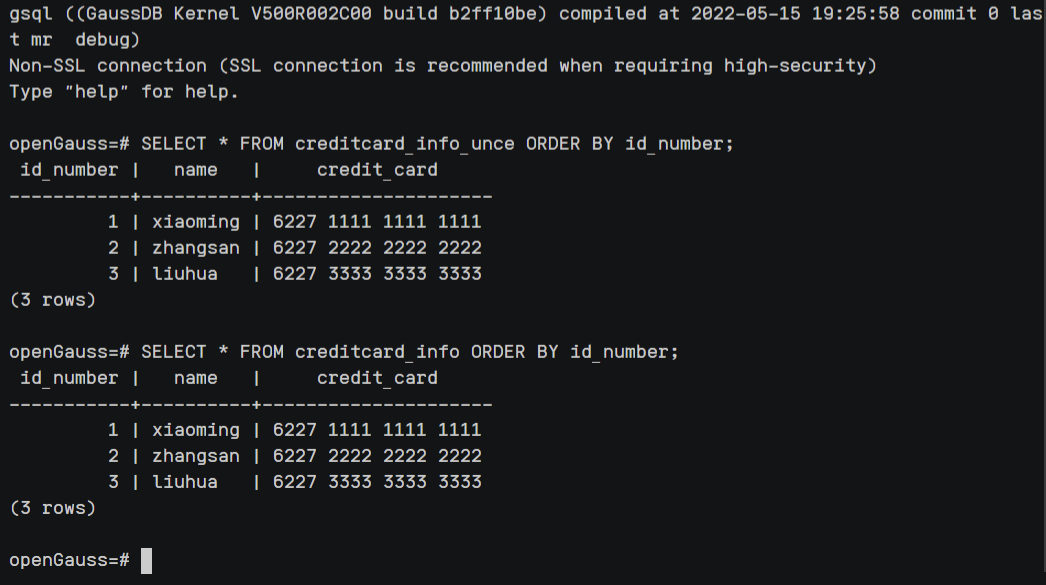
# 关卡四：openGauss密态数据库特性应用

任务一：物化视图的使用

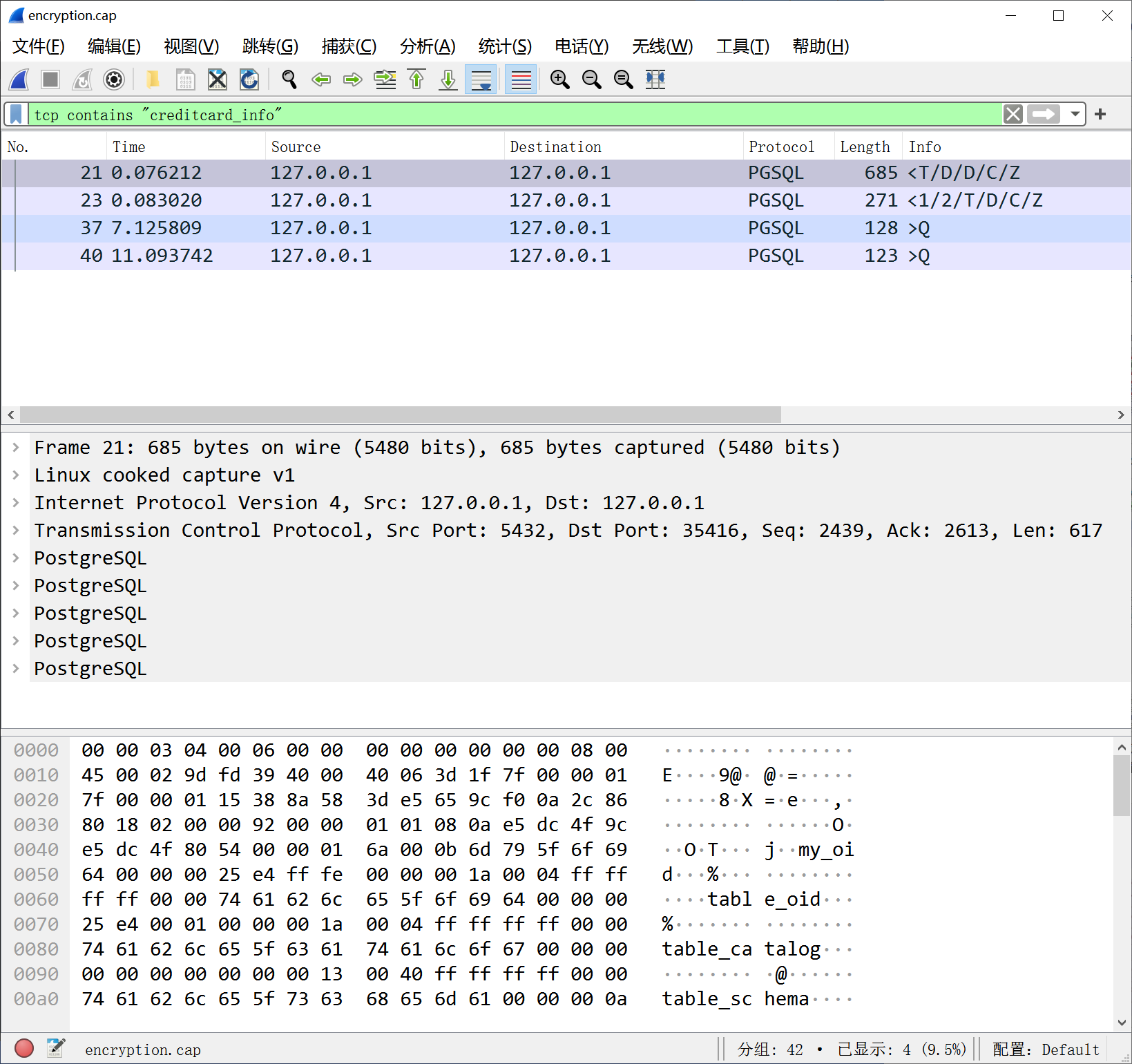
1. 通过tcpdump抓取数据流，此putty窗口暂时保持不动，将执行结果截图：

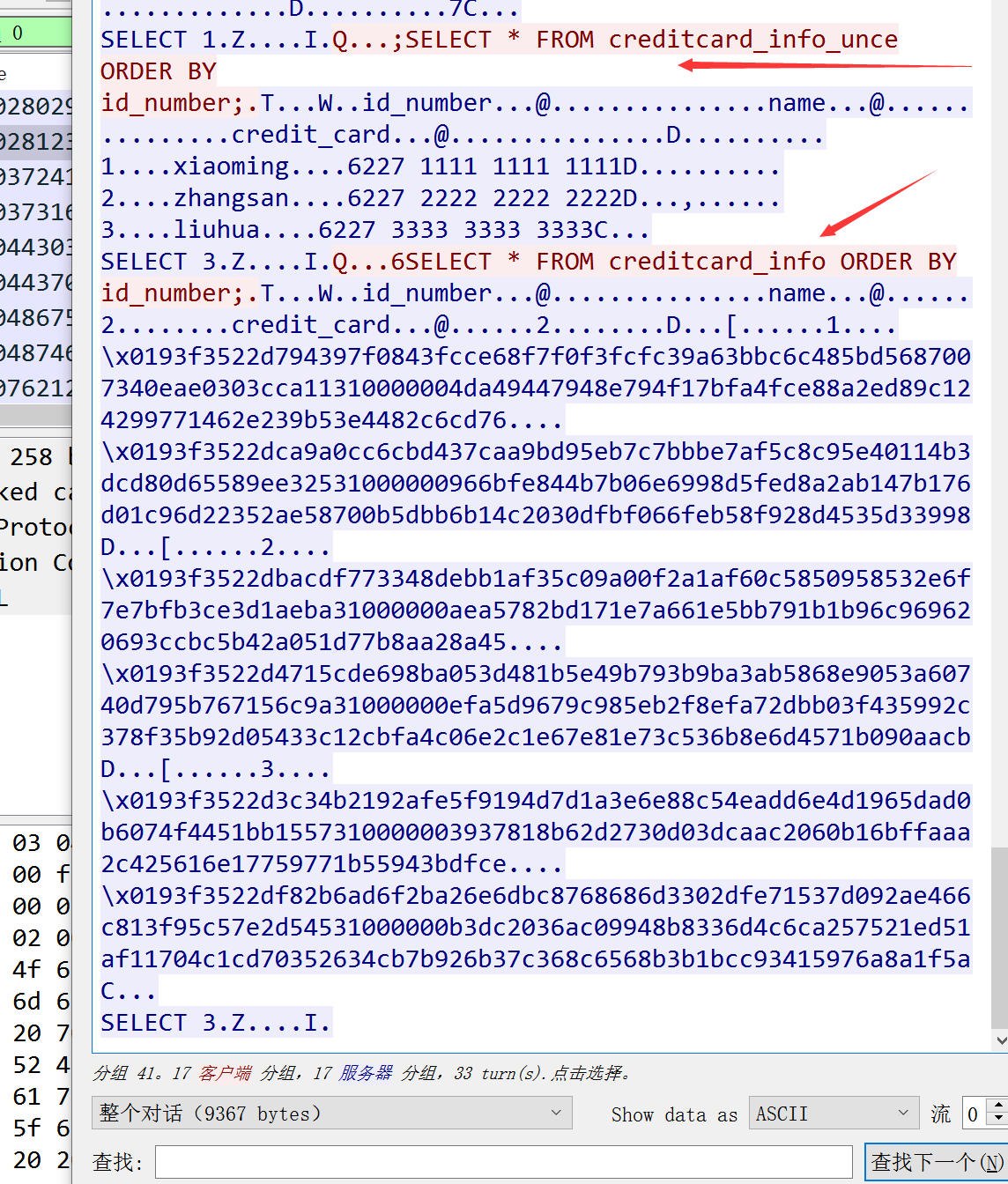


1. 将加密表和非加密表查询结果截图：

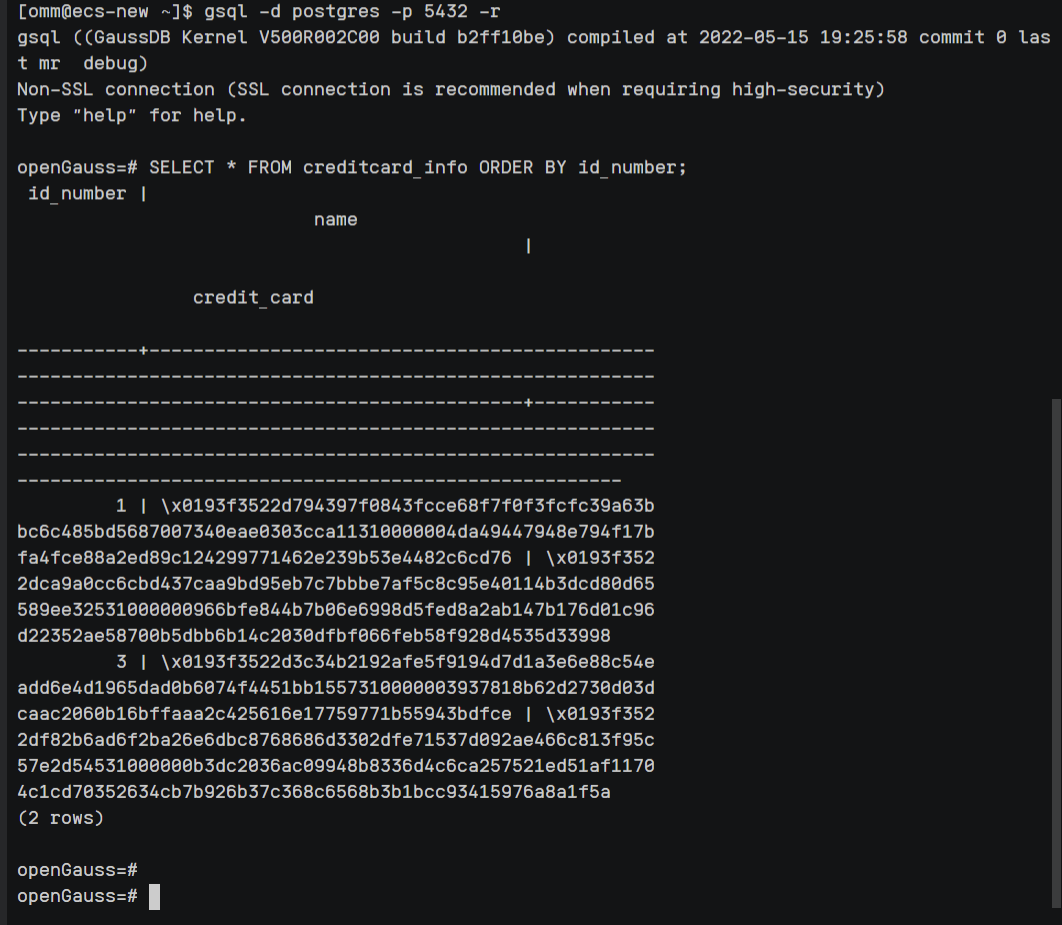


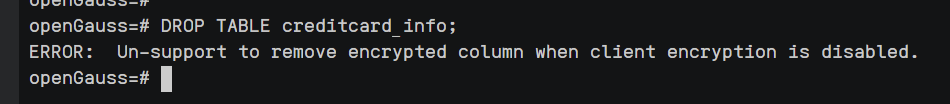
1. 用wireshark解析加密表和非加密表的差异时，非加密表name列和credit\_card列是明文，加密表name列和credit\_card列均是密文，将执行结果截图：





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：储存在物理磁盘上的应该是密文，这样可以更好的防止数据泄密。加密解密应该都在客户端，这样可以更好防止数据传输过程中泄密。