openGauss AI特性创新实践课



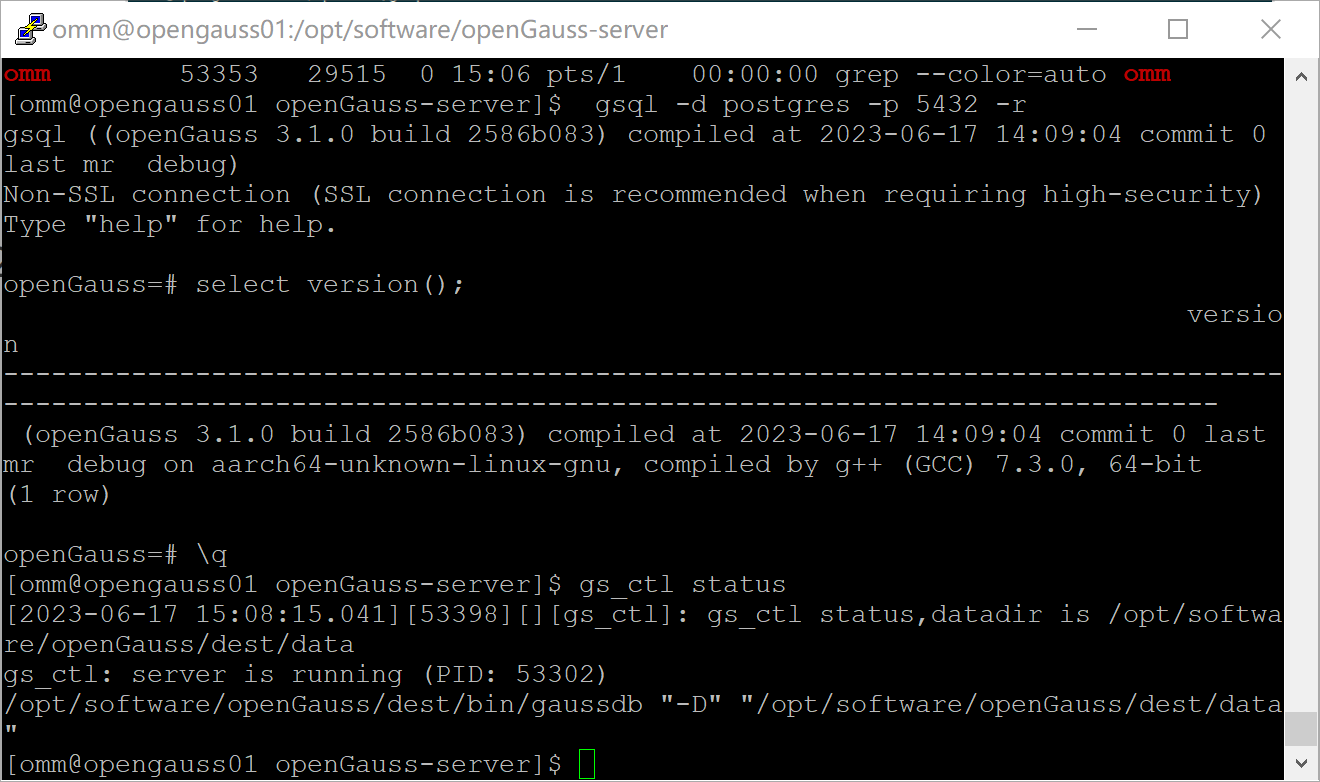
华为技术有限公司

# 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：

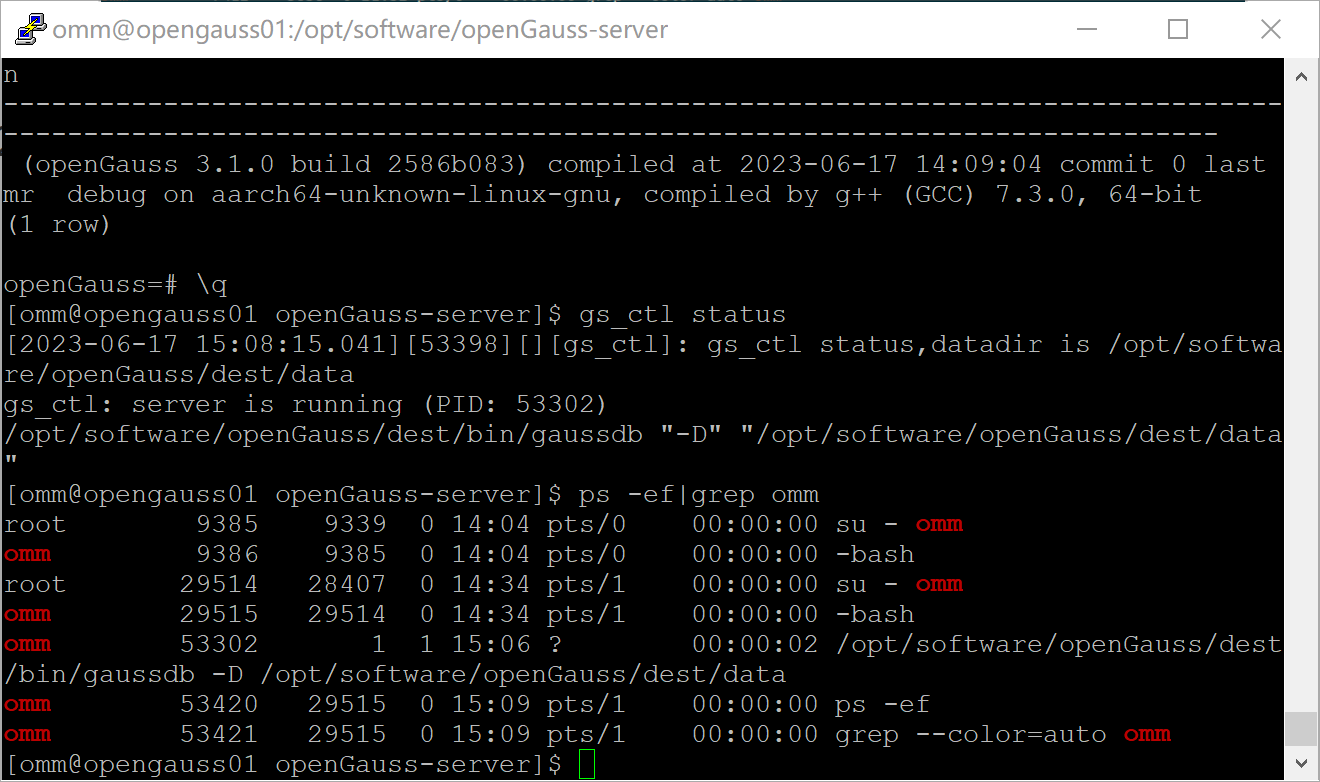
任务一：数据库状态验证

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）



实验思考题：为什么需要通过源码编译，安装数据库？

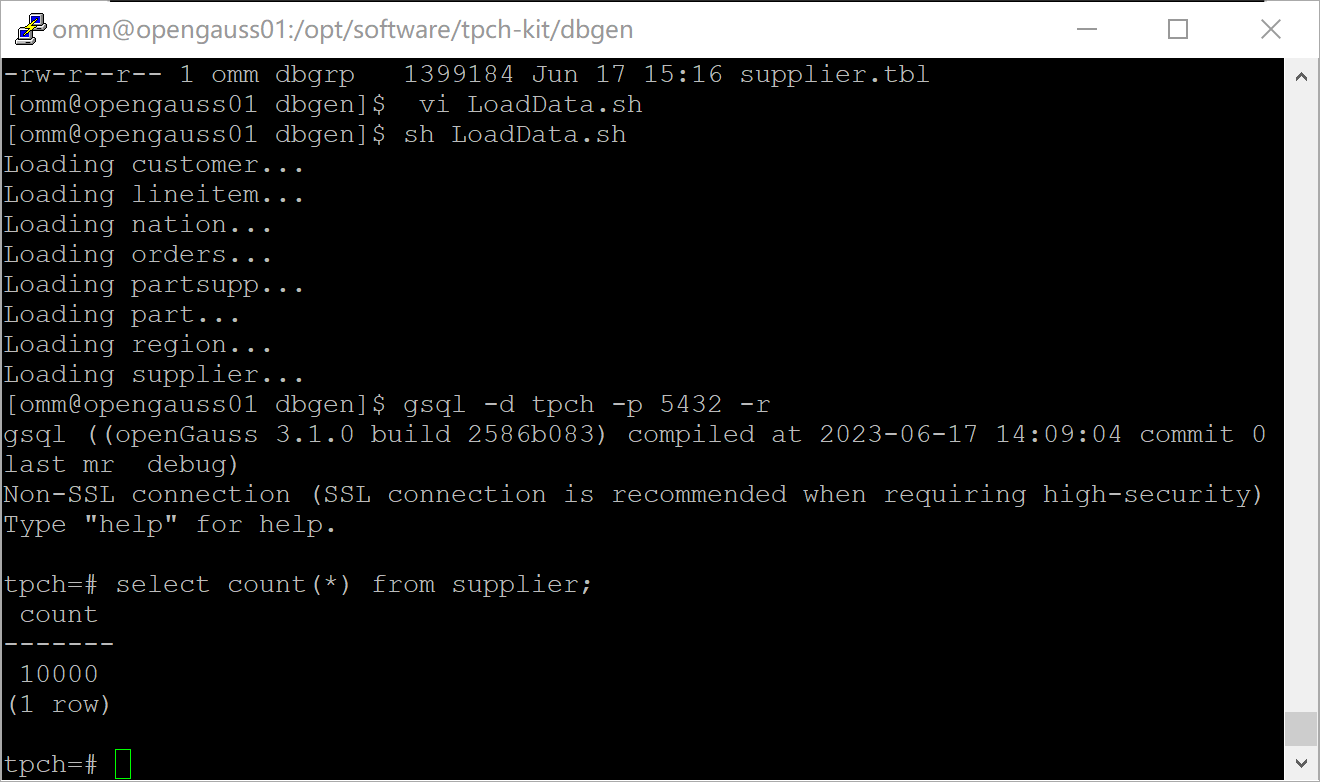
答：可以灵活设定参数，按照需求进行安装，可以自己选择安装的版本和安装目录。

# 关卡二、openGauss数据导入及基本操作

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：

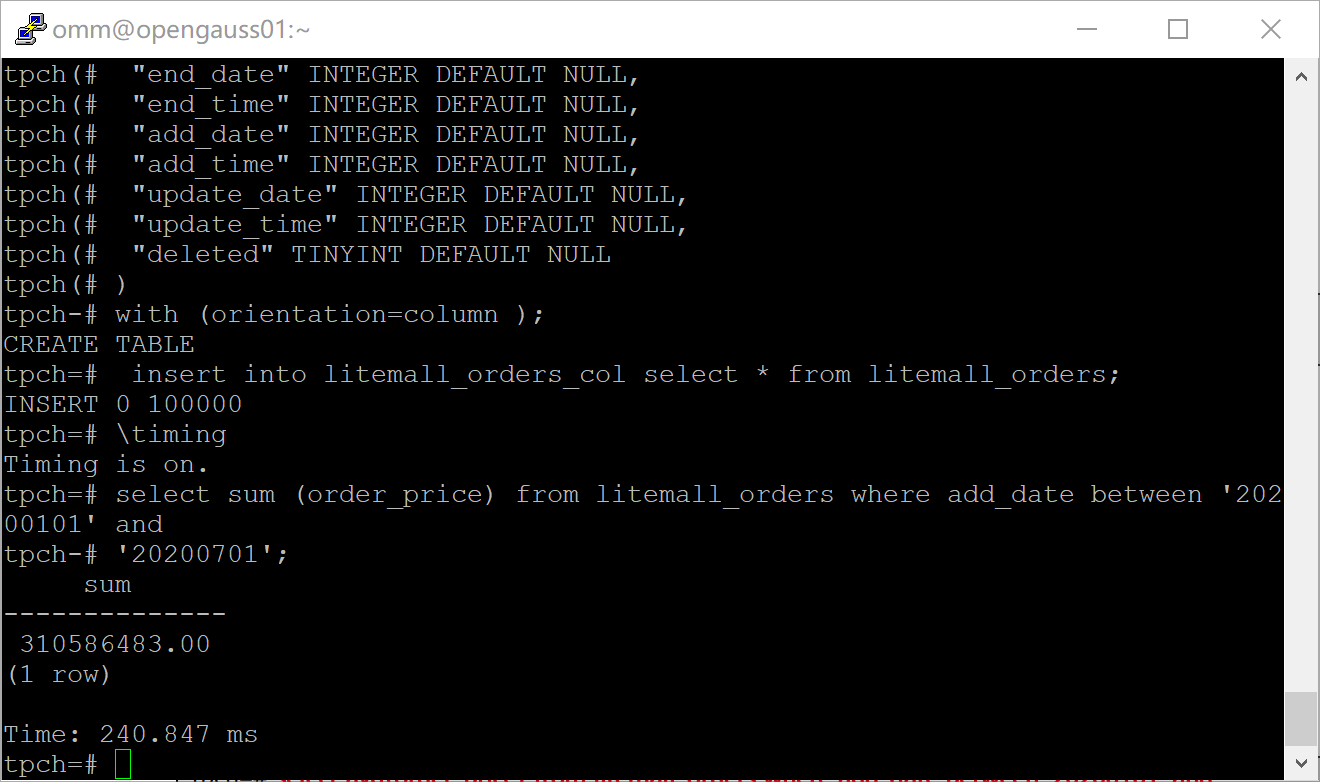
select count(\*) from supplier;;



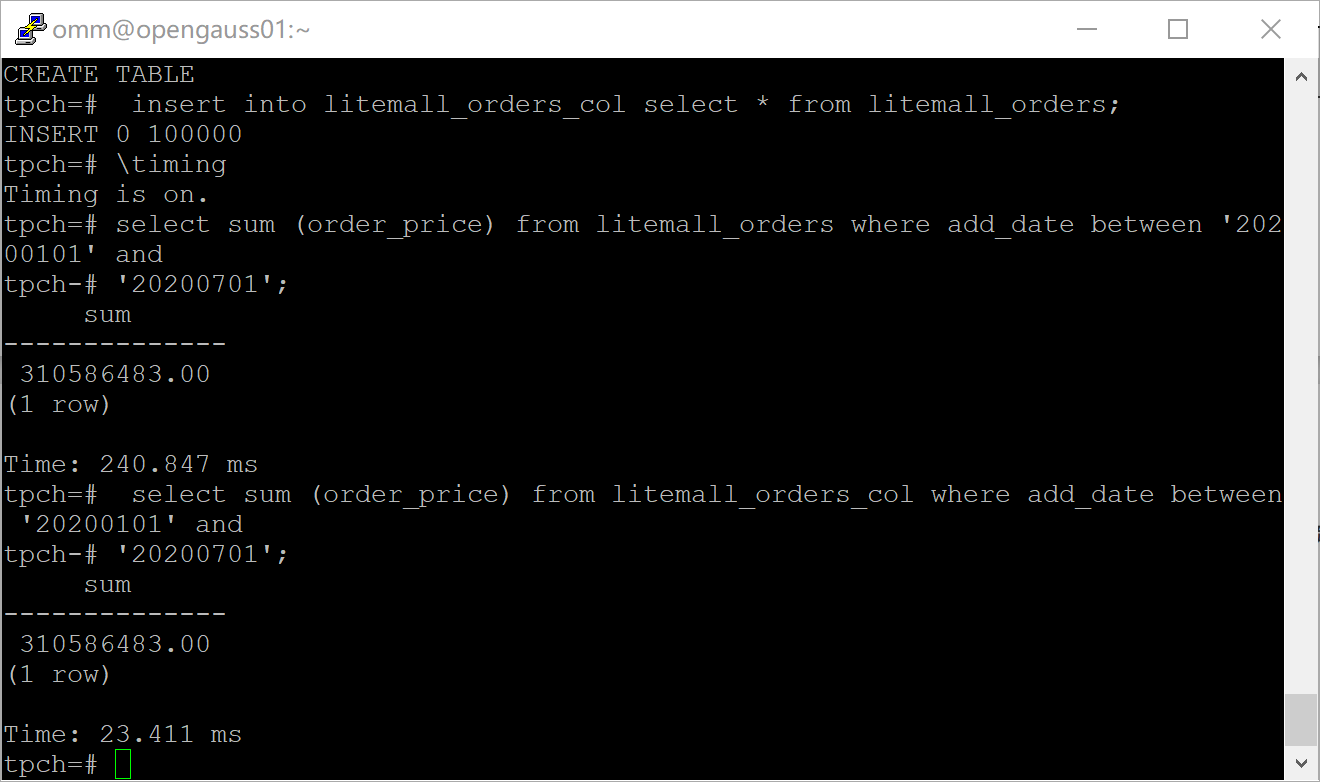
任务二：行存表与列存表执行效率对比

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';

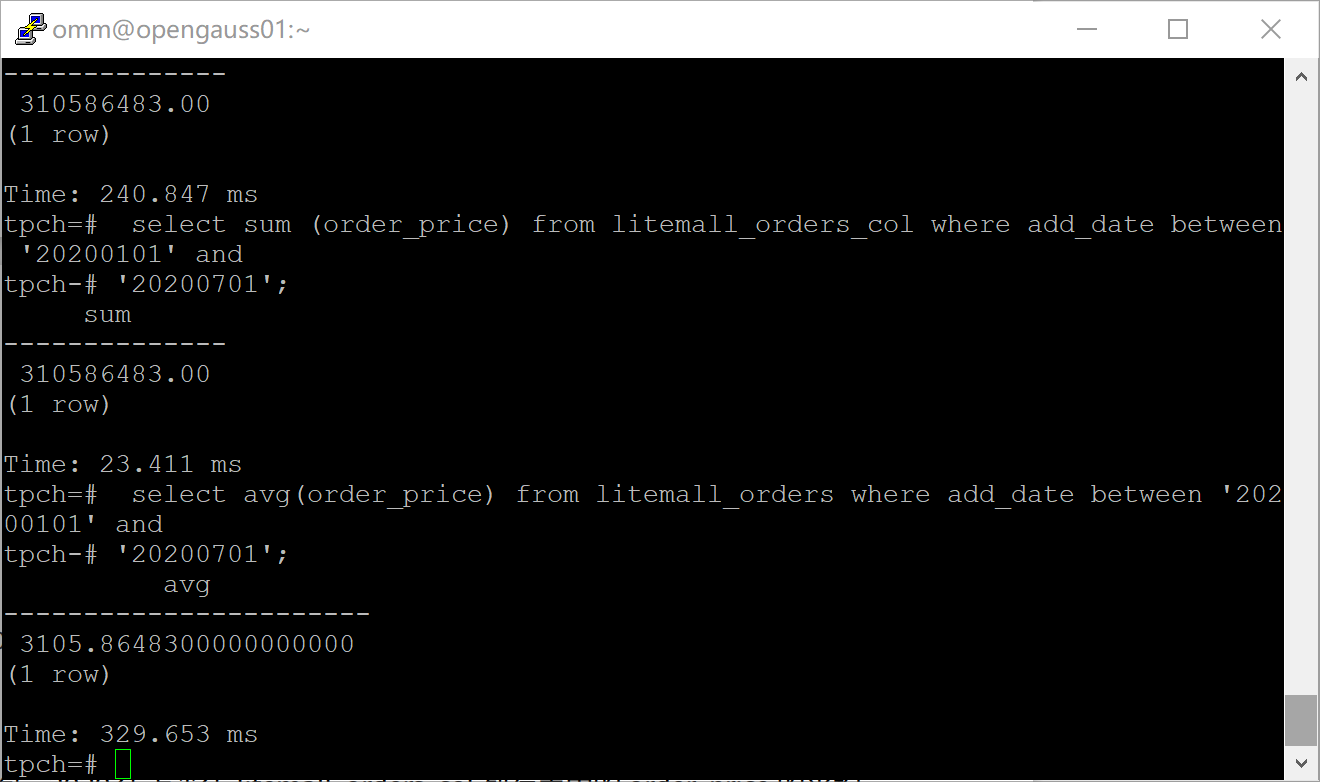


select sum (order\_price) from litemall\_orders\_col where add\_date between '20200101' and '20200701';

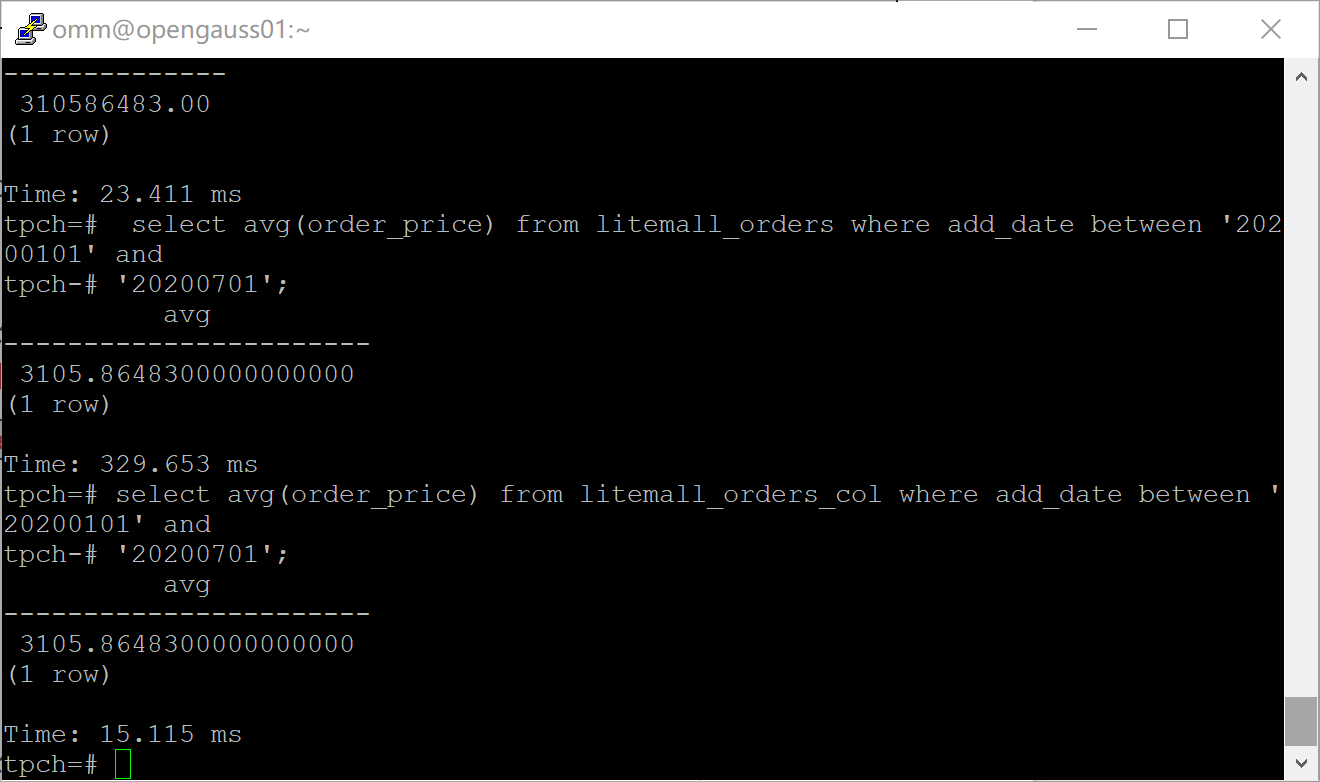


2. 2020年上半年litemall\_orders行存表与litemall\_orders\_col列存表中的order\_price的平均值查询，并对比执行效率截图

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

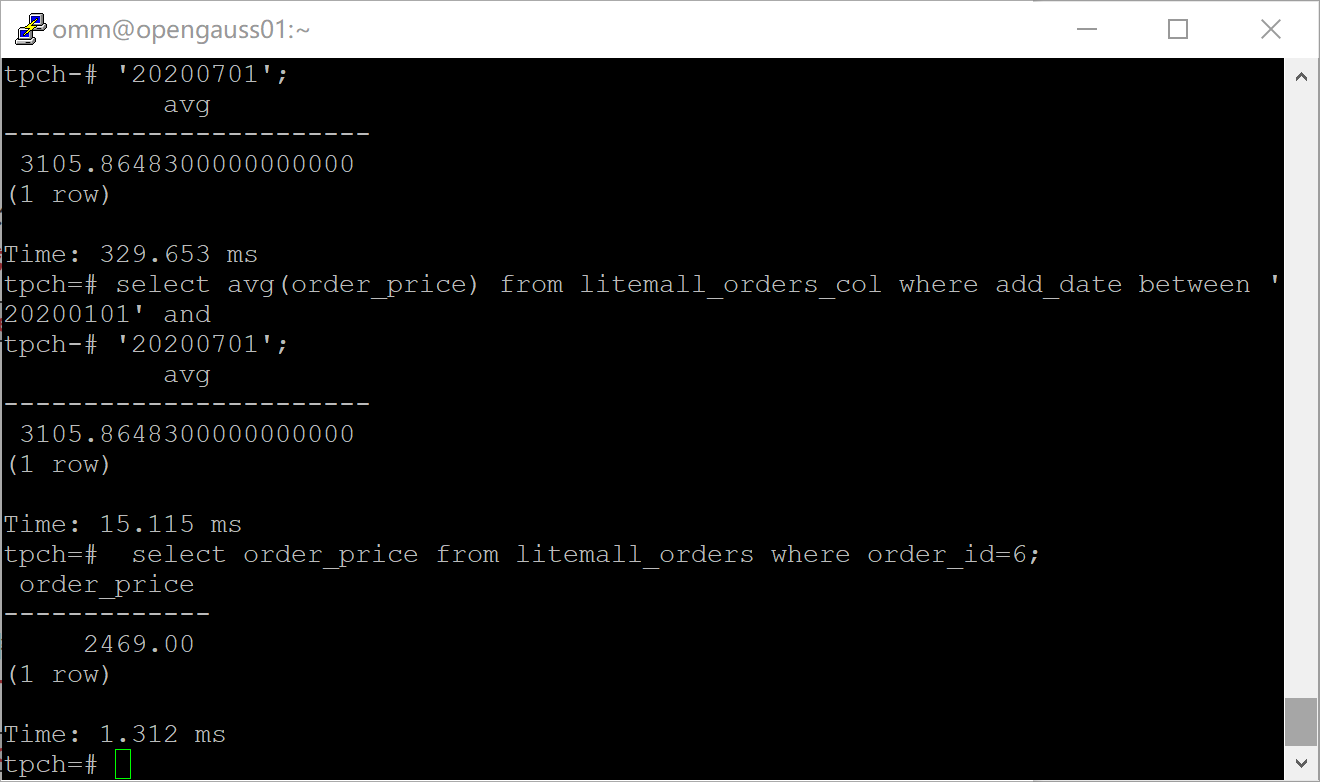


select avg (order\_price) from litemall\_orders\_col where add\_date between '20200101' and '20200701';

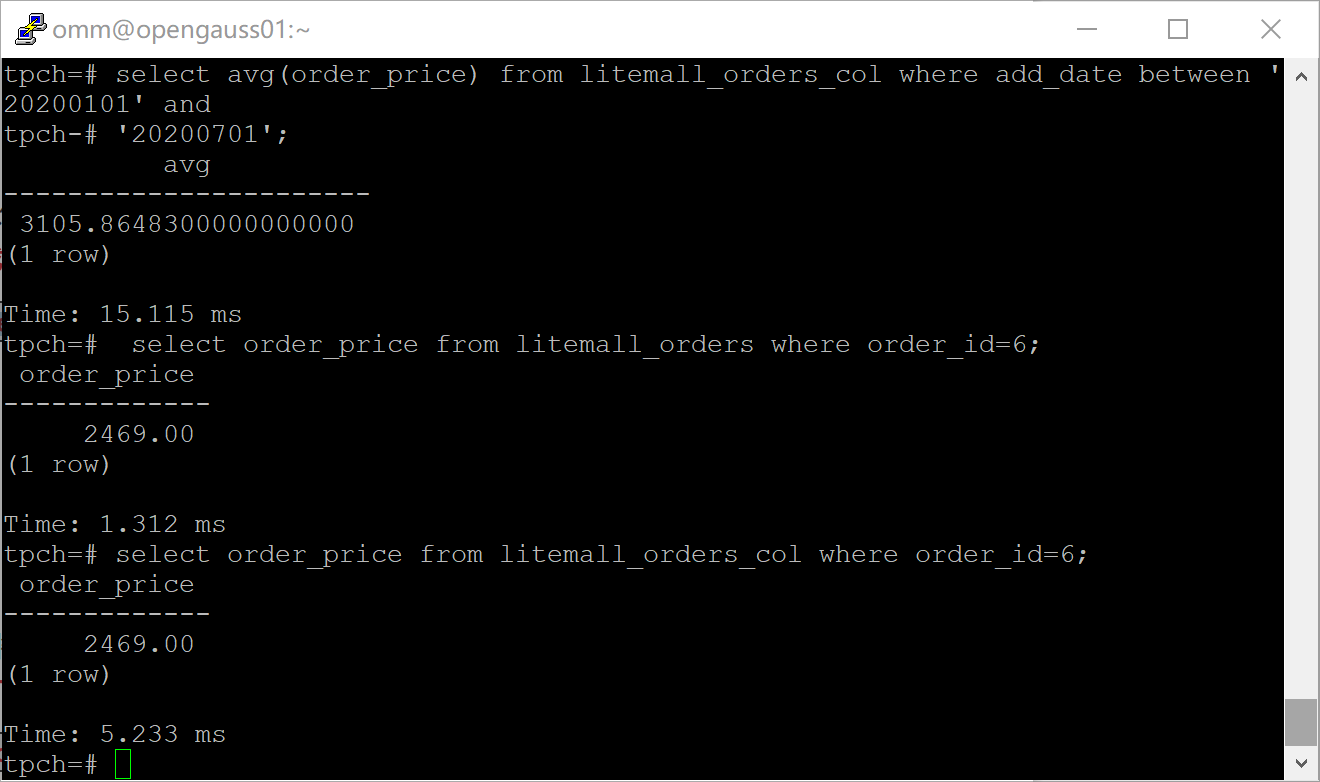


3. 查询litemall\_orders行存表与litemall\_orders\_col列存表中order\_id为6的order\_price的值，并对比执行效率截图。

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

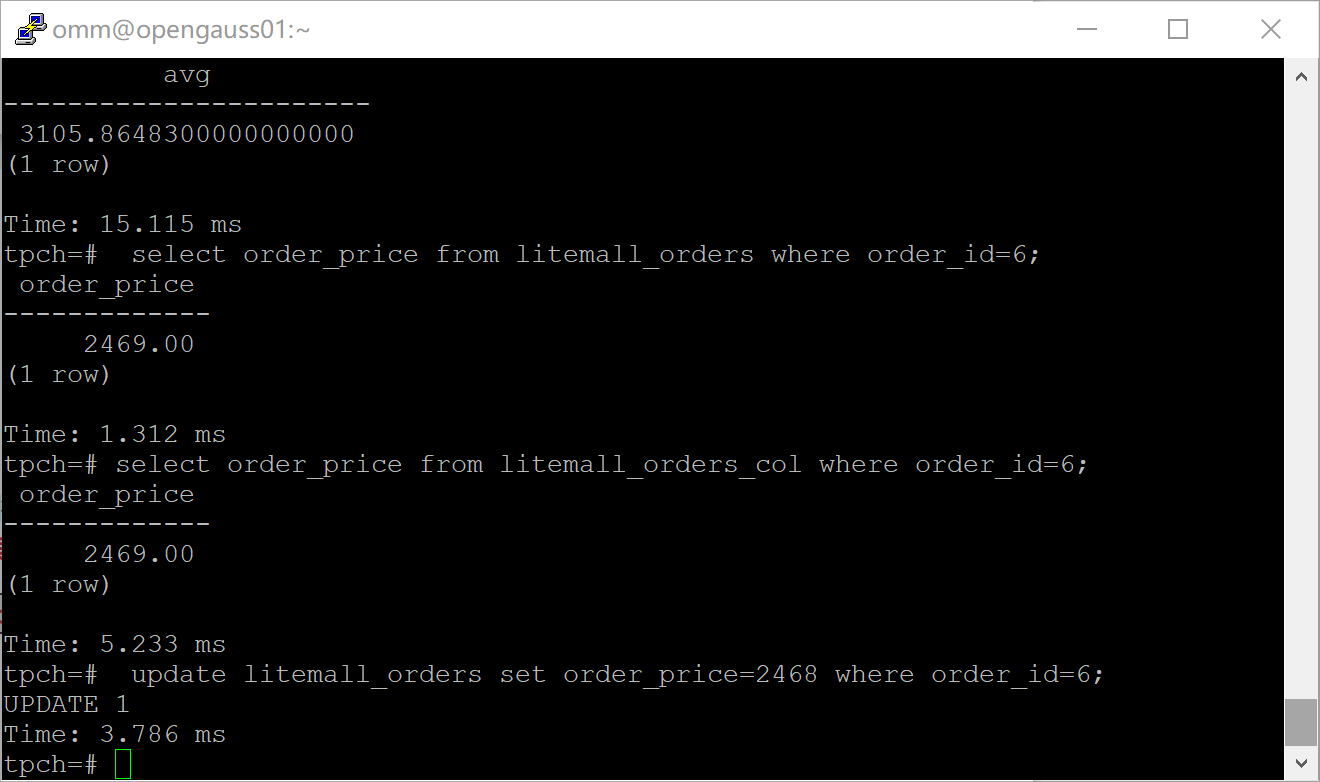


select order\_price from litemall\_orders\_col where order\_id=6;

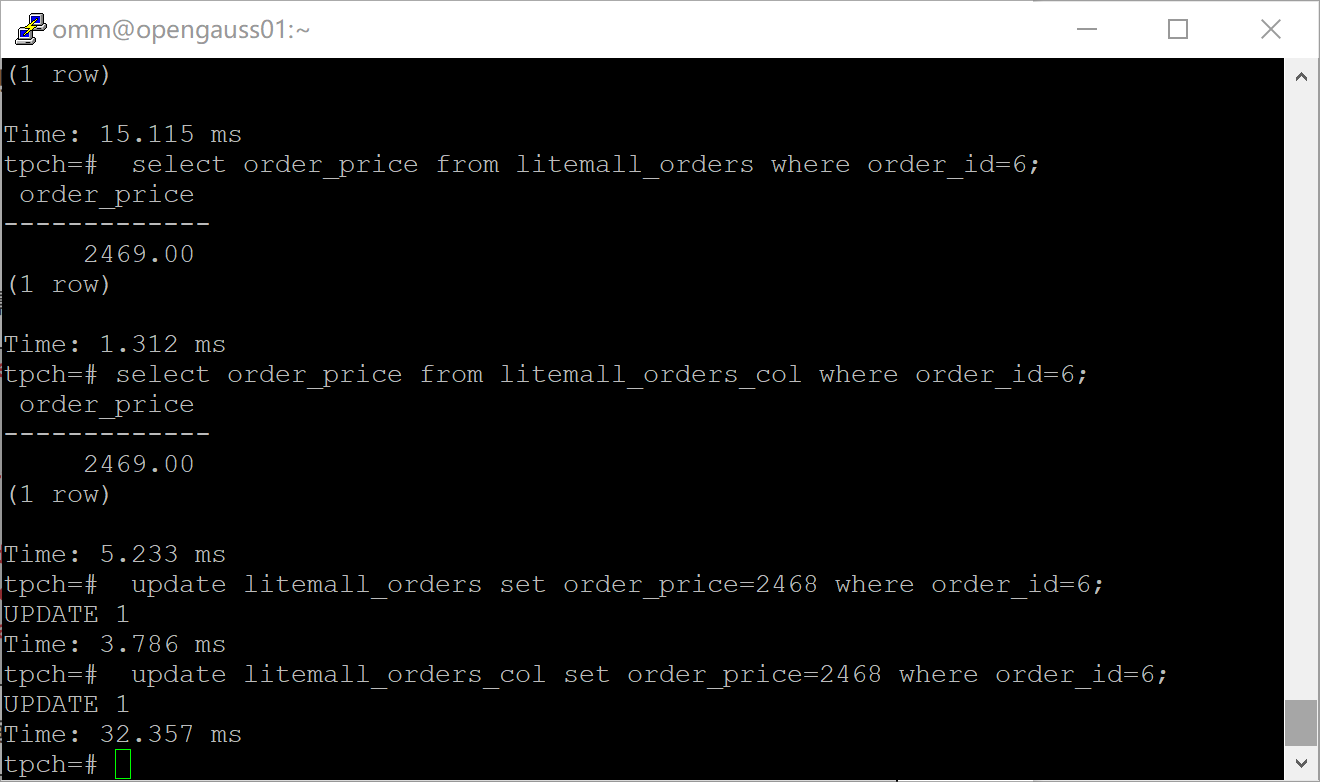


4. 将litemall\_orders行存表与litemall\_orders\_col列存表中order\_id为6的order\_price修改为2468，并对比执行效率截图。

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



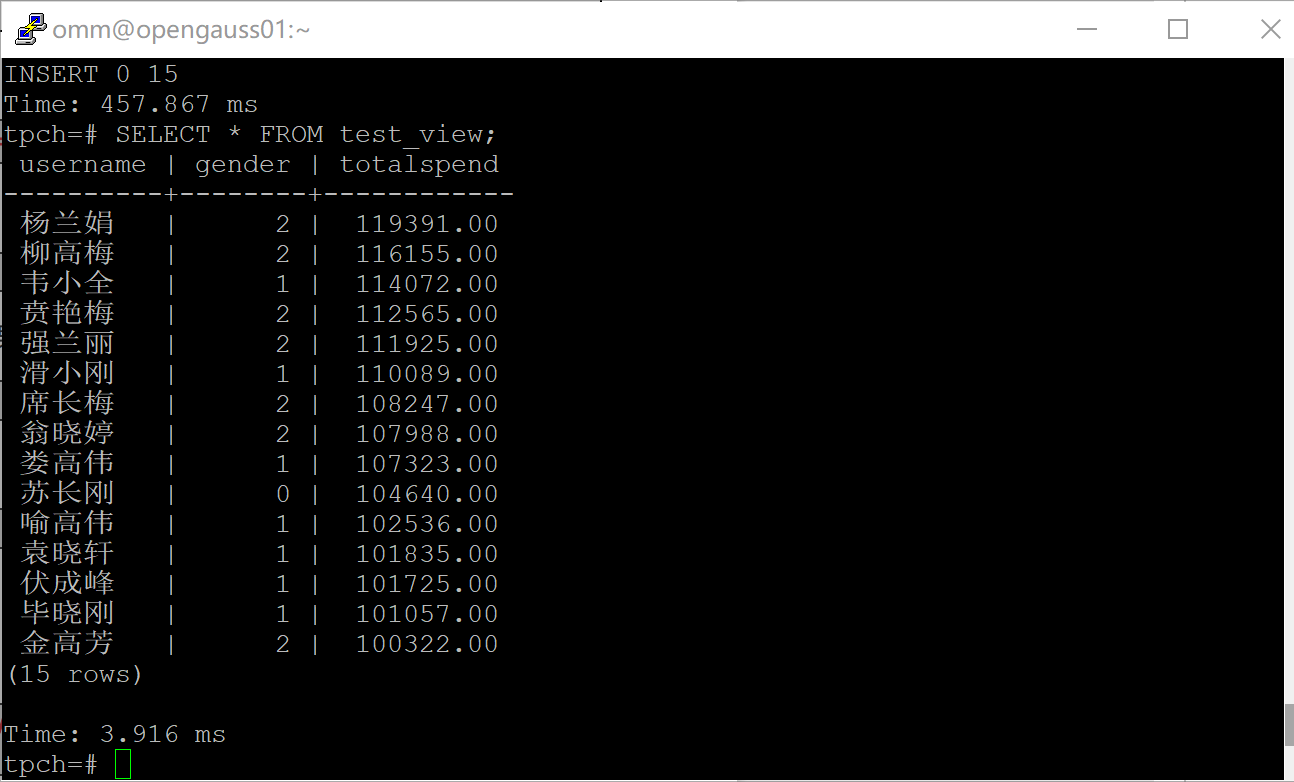
update litemall\_orders\_col set order\_price=2468 where order\_id=6;



任务三：物化视图的使用

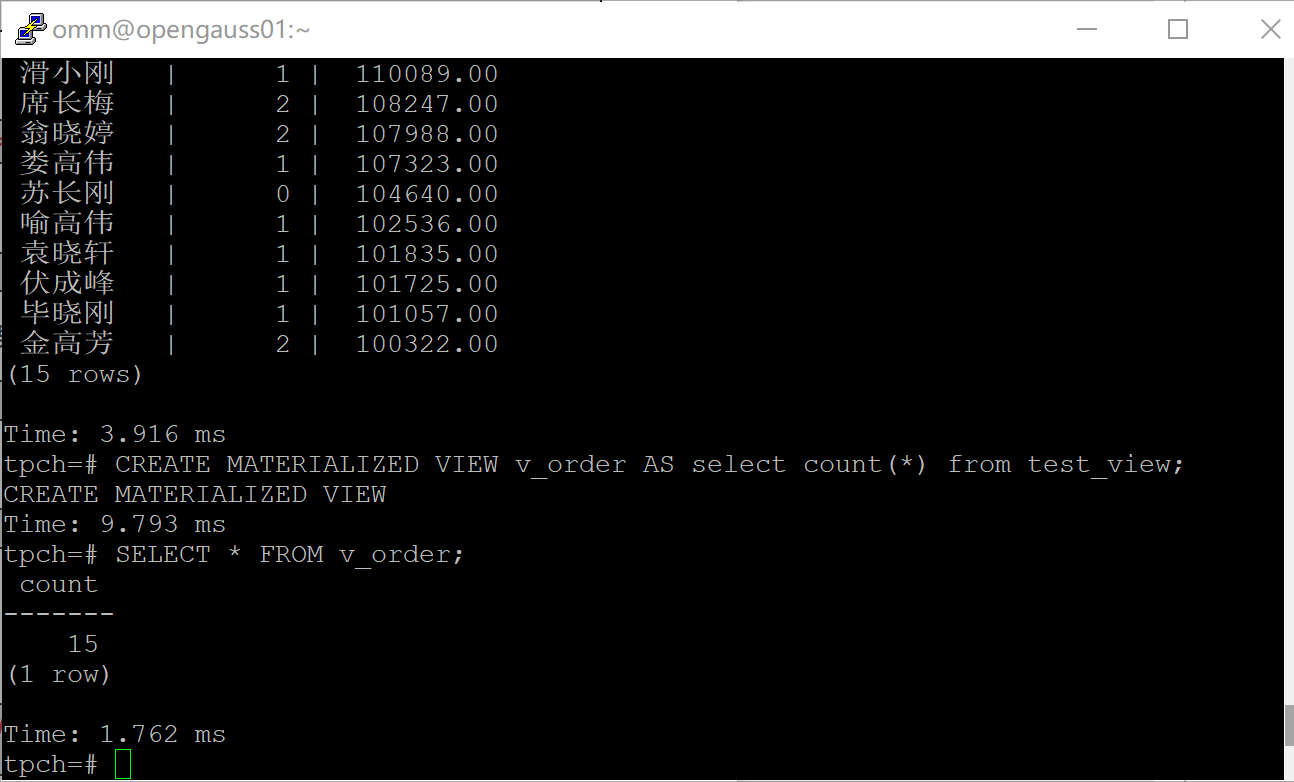
1. 创建物化视图所需要的表后，对表内容进行查询，对查询结果截图：

SELECT \* FROM test\_view;



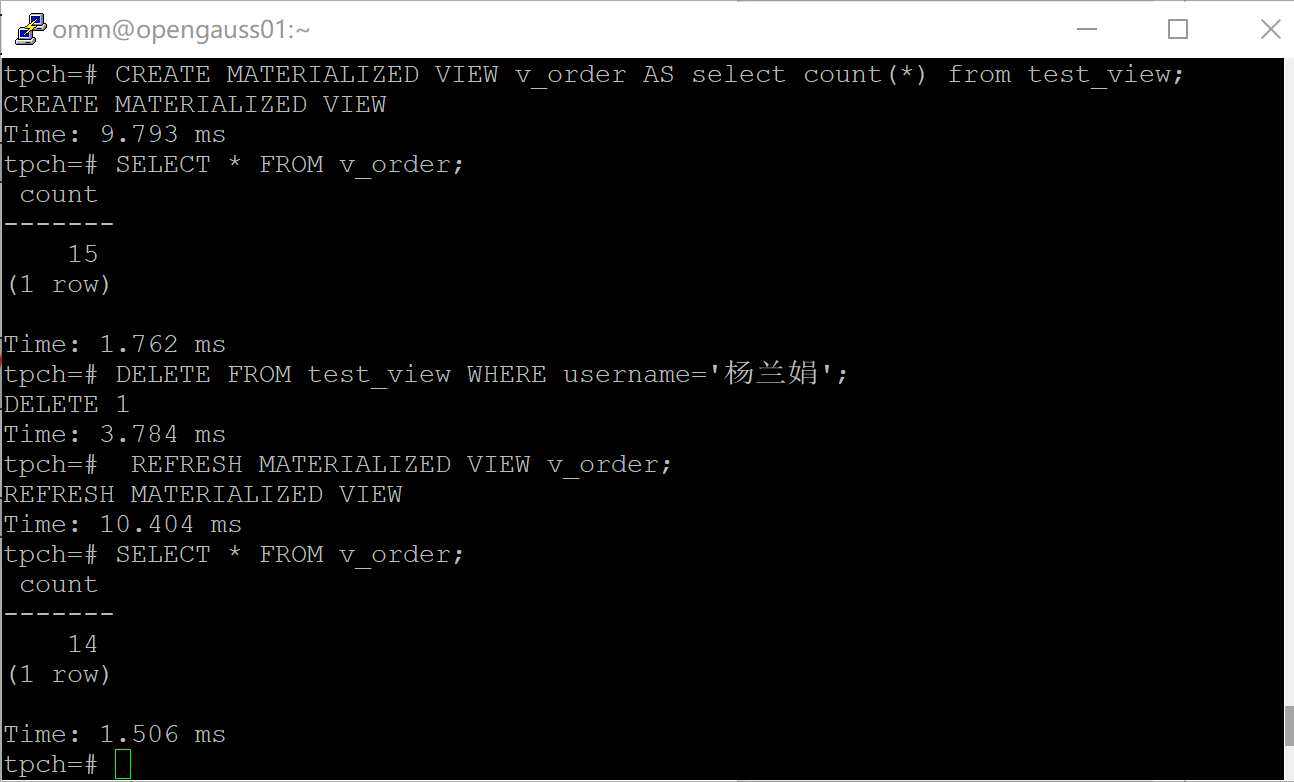
2. 使用物化视图统计人数，查询物化视图结果，将执行结果截图。

SELECT \* FROM v\_order;



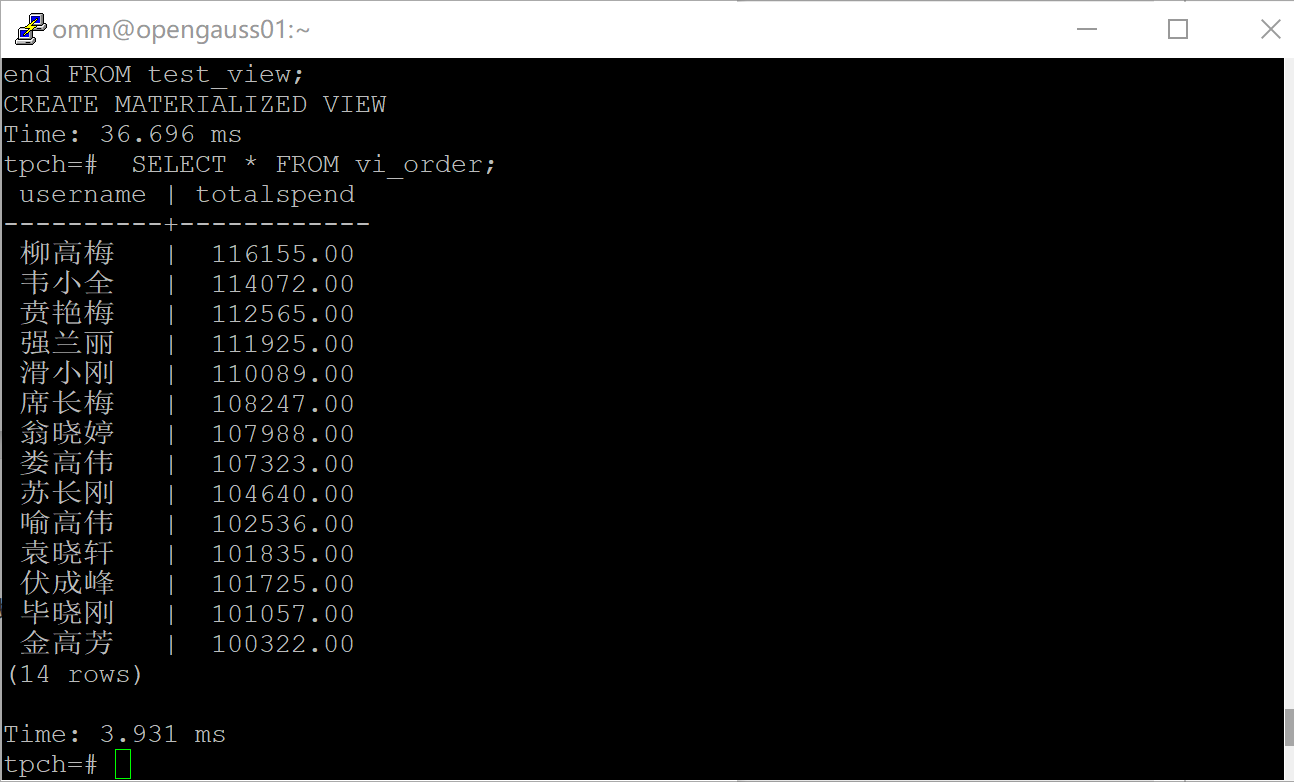
3. 对表进行操作后，刷新物化视图，查询物化视图结果，将执行结果截图。

SELECT \* FROM v\_order;



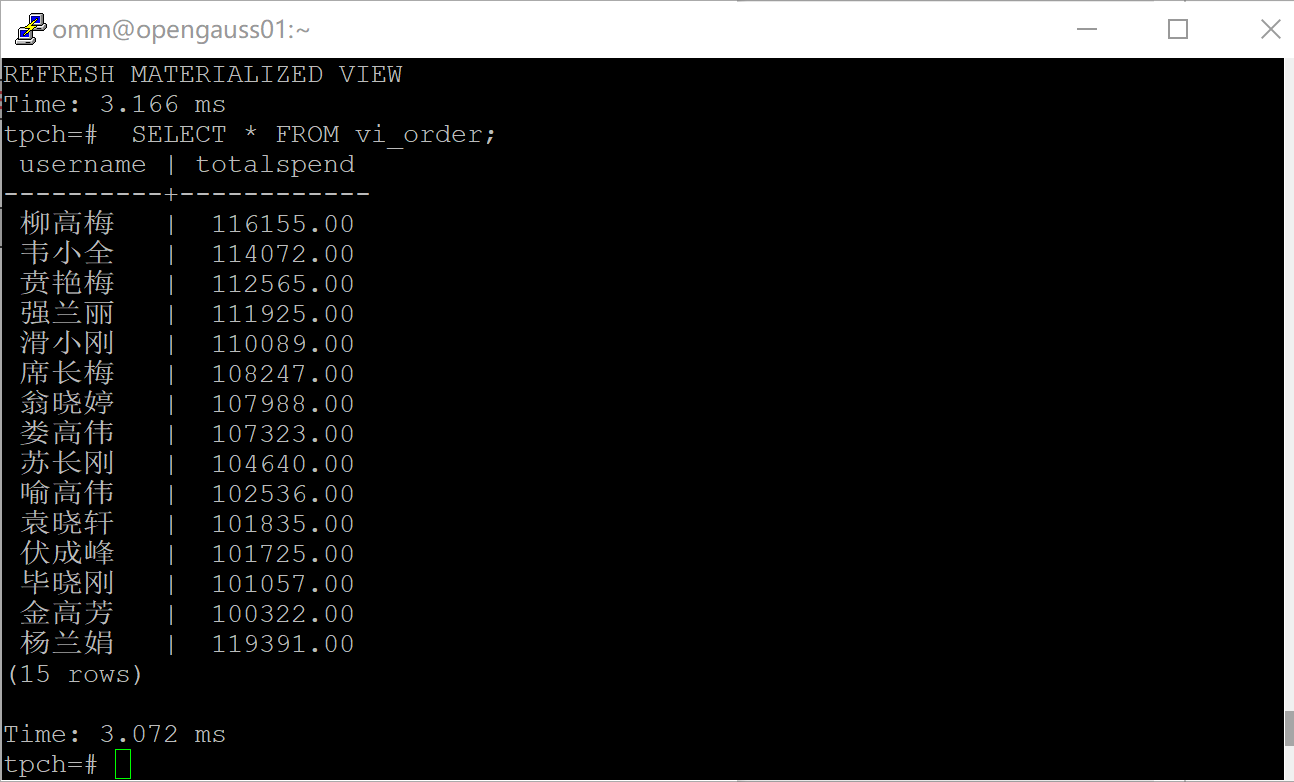
4. 创建增量物化视图，查询物化视图结果，将执行结果截图。

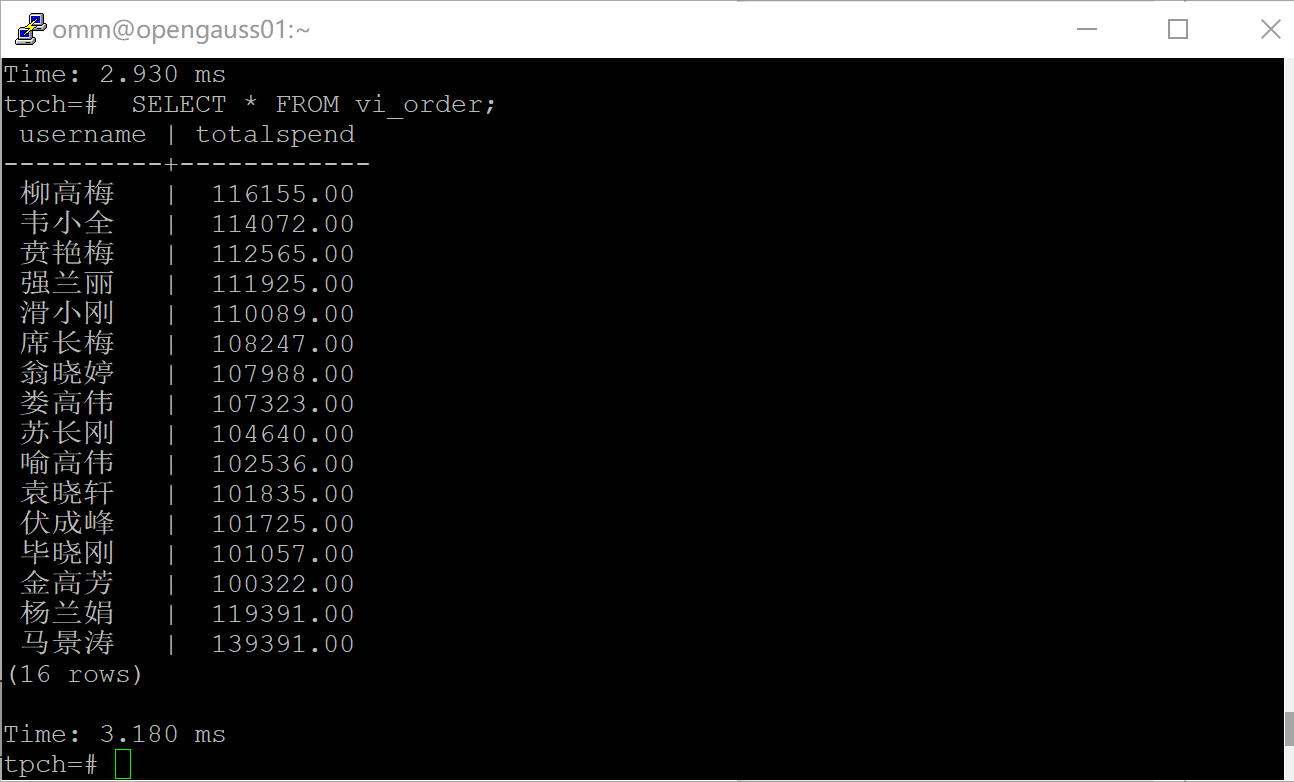
SELECT \* FROM vi\_order;



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;





实践思考题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？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：（1）行存表按行存储数据，而列存表按列存储数据。这导致在执行查询时需要读取的数据量不同。行存表需要读取整行数据，而列存表只需读取所需列的数据。如果查询需要读取的数据量较大，列存表的效率会更高，因为只需读取所需列的数据而无需读取其他列的数据。

（2）执行单条记录的查询或修改操作时，行存表效率更高。

（3）执行大规模聚合查询、数据统计和分组操作时，列存表效率更高。

实践思考题2：全量物化视图与增量物化视图有哪些差别？

答：（1）刷新方式：全量物化视图在刷新时重新计算和加载整个视图的数据，通常需要执行完整的数据抽取、转换和加载过程；增量物化视图只更新或加载源数据发生变化的部分，而不是重新计算和加载整个视图的数据。

（2）存储和性能：全量物化视图存储了视图的完整数据集，占用较大的存储空间，它的刷新时间较长，但查询时性能通常较好；增量物化视图只存储源数据的部分或计算结果，通常占用较小的存储空间，刷新时间较短，但在查询时可能需要进行额外的计算或合并操作，因此查询性能可能受到一定影响。

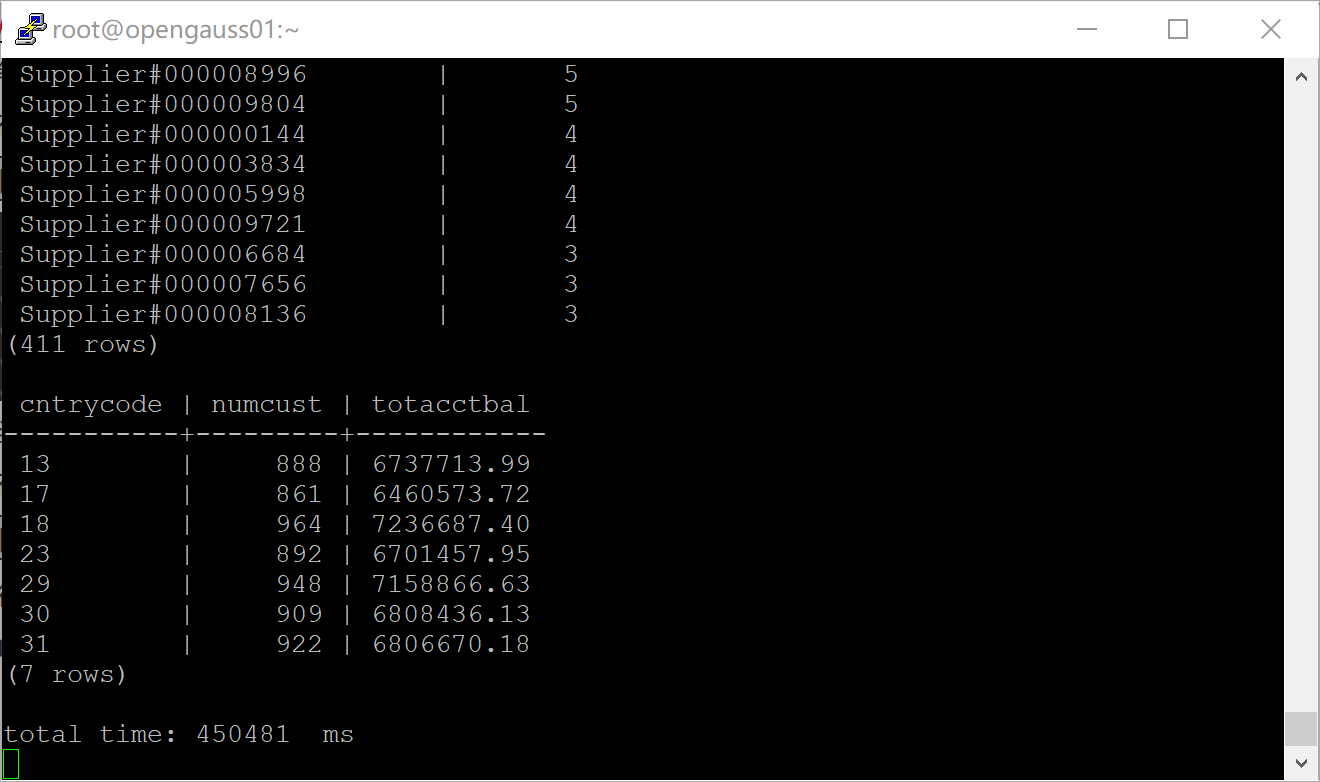
（3）更新方式：全量物化视图：全量物化视图在刷新时会删除旧数据，重新计算和加载新数据；增量物化视图通过增量更新方式，只处理源数据的变化部分，不需要删除和重新加载整个视图数据。

# 关卡三、openGauss的AI4DB特性应用

任务一：使用X-Tuner进行参数优化

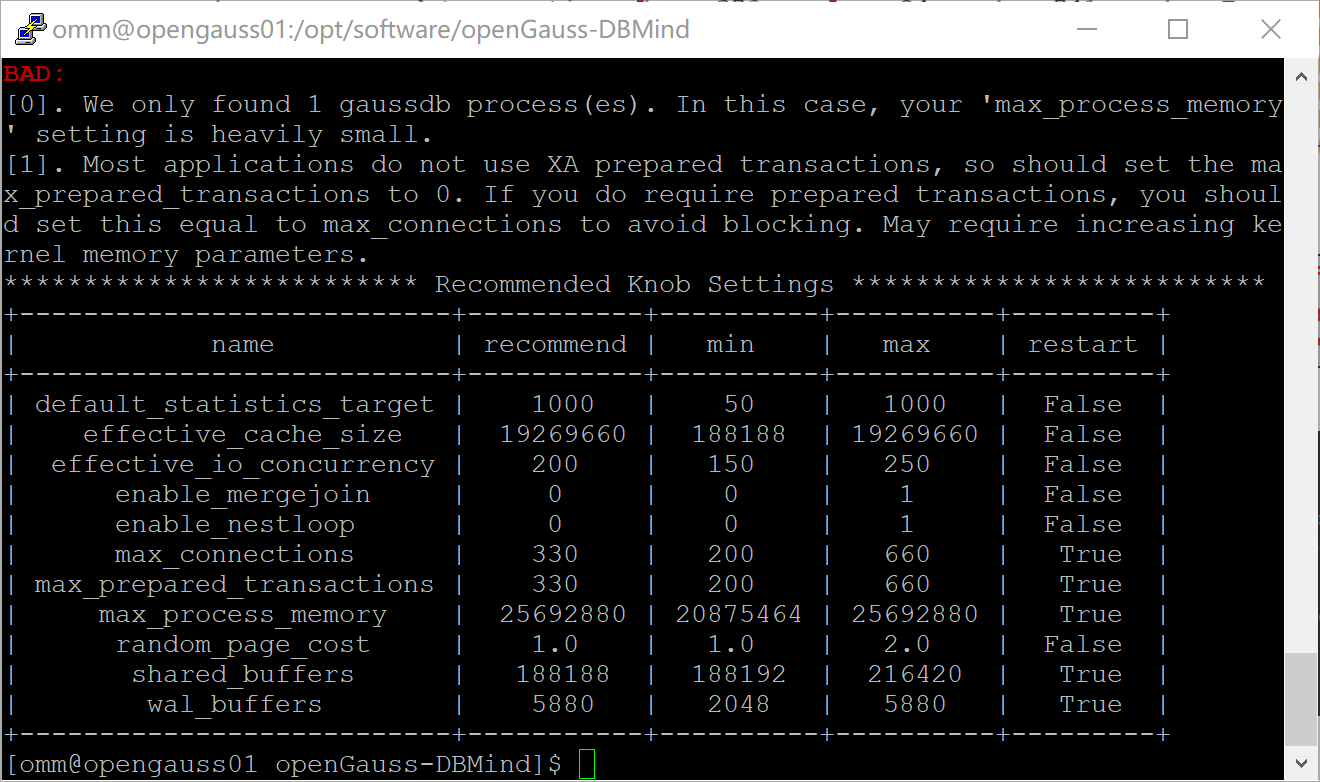
1. 执行TPCH脚本，获得测试时间，将执行结果截图：

gsql -d tpch -p 5432 -r -f /opt/software/tpch-kit/dbgen/queries/queries.sql > /opt/software/tpch-kit/dbgen/queries/queries01.log



2. 使用root用户，执行X-Tuner进行参数建议优化，将执行结果截图

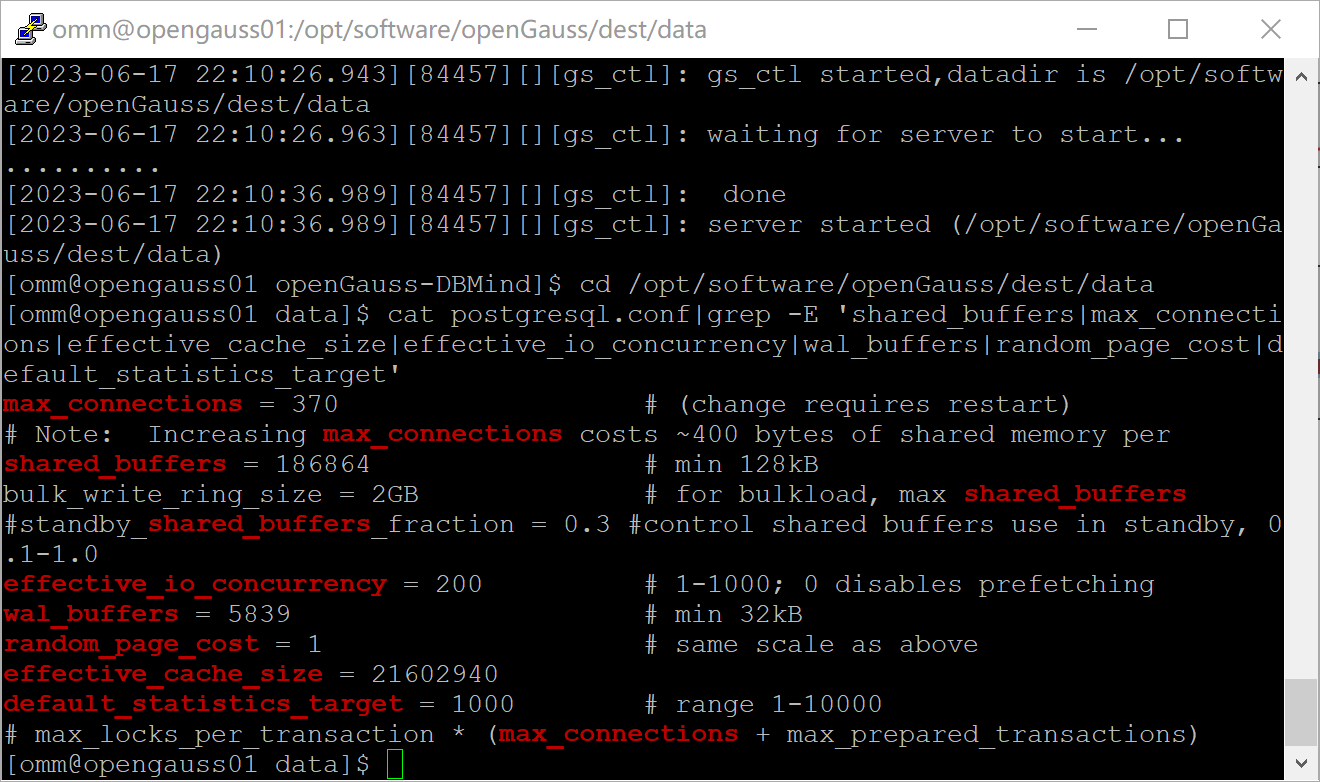
gs\_xtuner recommend --db-name tpch --db-user omm --port 5432 --host 127.0.0.1 --host-user omm



3.重启完成后，获取参数值：

cd /opt/software/openGauss/data

cat postgresql.conf|grep -E 'shared\_buffers|max\_connections|effective\_cache\_size|effective\_io\_concurrency|wal\_buffers|random\_page\_cost|default\_statistics\_target'



任务二：使用Index-advisor对select 查询语句进行优化，并通过对比执行计划，得到优化前后的不同。

1. 使用explain，对查询2020年3月订单表收入并进行排序的SQL加以分析，将结果截图。

EXPLAIN

SELECT ad.province AS province, SUM(o.actual\_price) AS GMV

FROM litemall\_orders o,

address\_dimension ad,

date\_dimension dd

WHERE o.address\_key = ad.address\_key

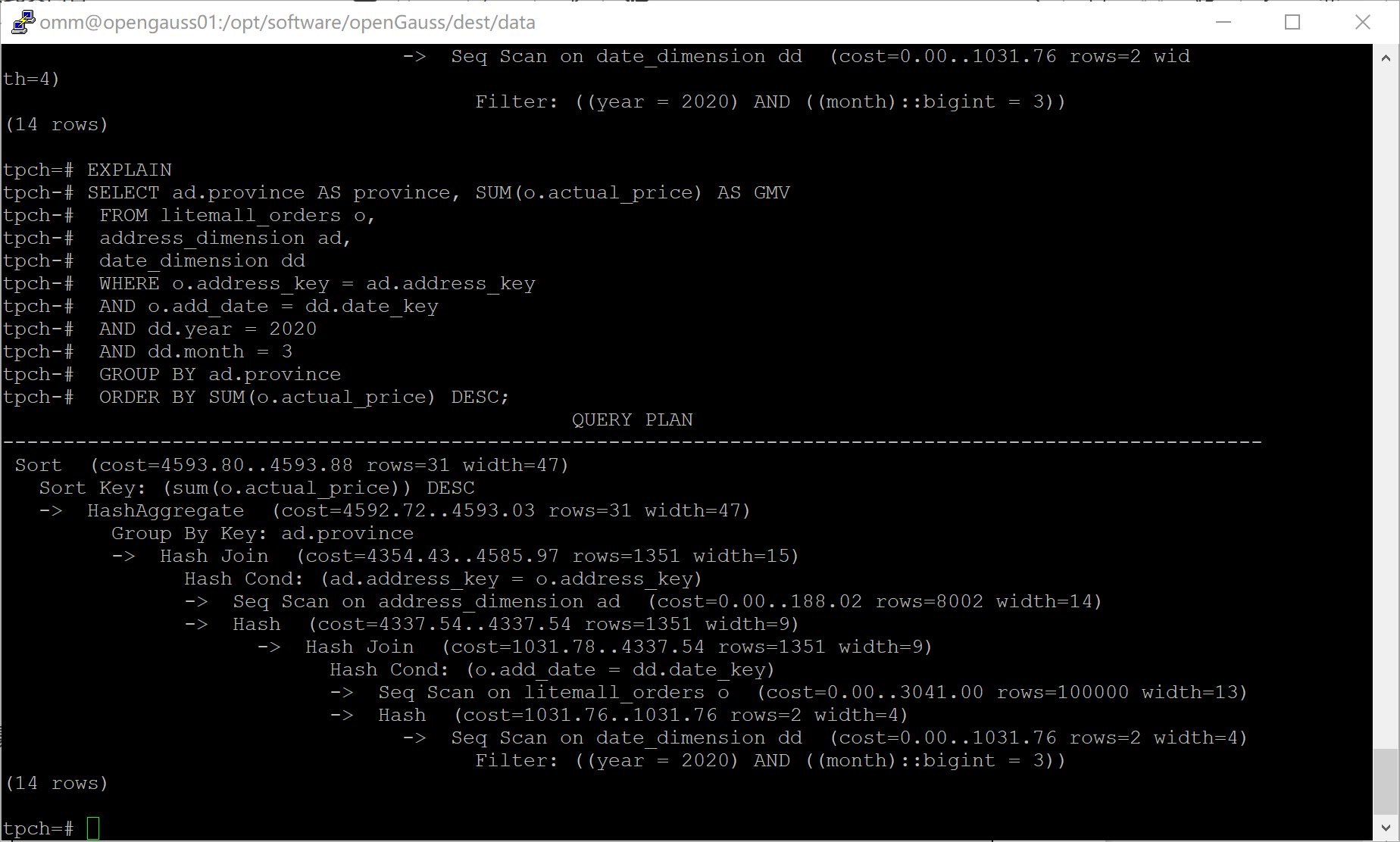
AND o.add\_date = dd.date\_key

AND dd.year = 2020

AND dd.month = 3

GROUP BY ad.province

ORDER BY SUM(o.actual\_price) DESC;



2. 使用索引推荐功能，对查询语句进行推荐，将执行结果截图。

select \* from gs\_index\_advise('

SELECT ad.province AS province, SUM(o.actual\_price) AS GMV

FROM litemall\_orders o,

address\_dimension ad,

date\_dimension dd

WHERE o.address\_key = ad.address\_key

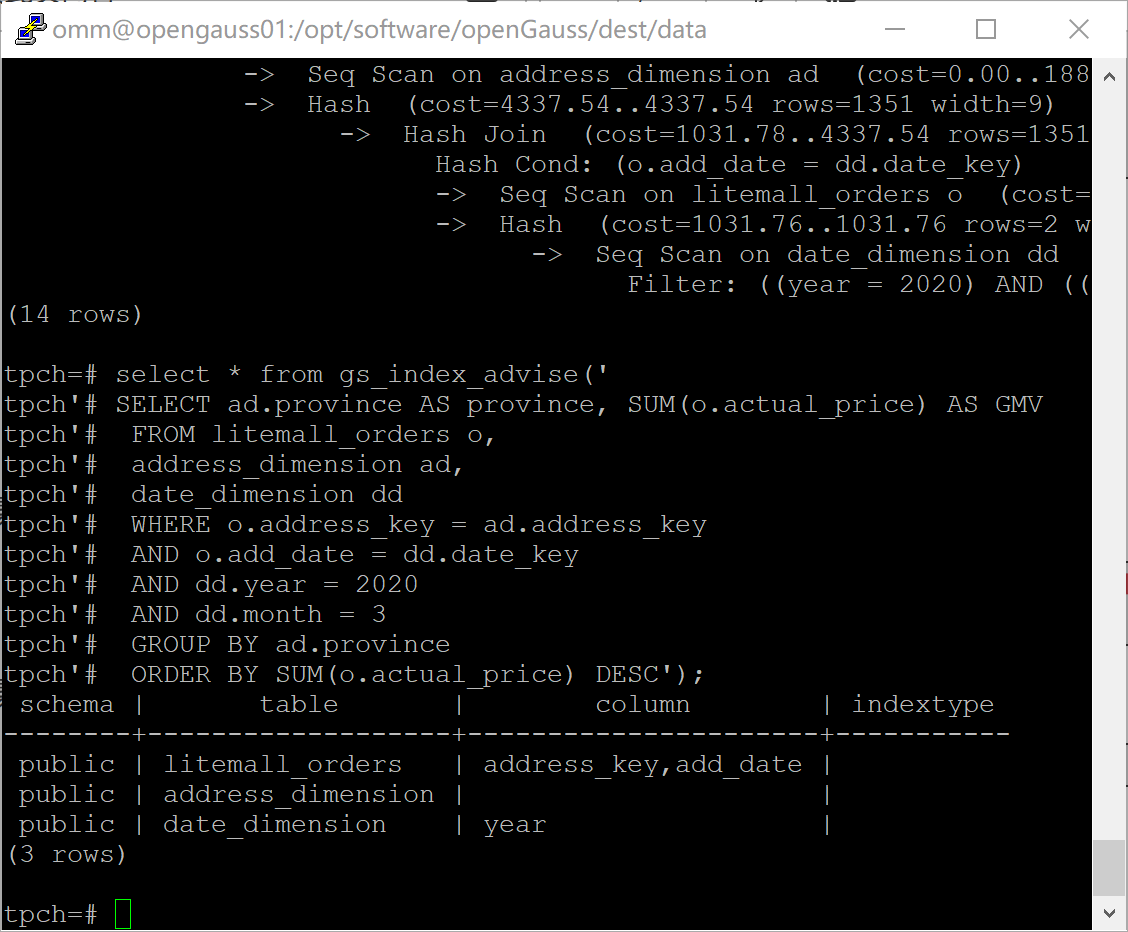
AND o.add\_date = dd.date\_key

AND dd.year = 2020

AND dd.month = 3

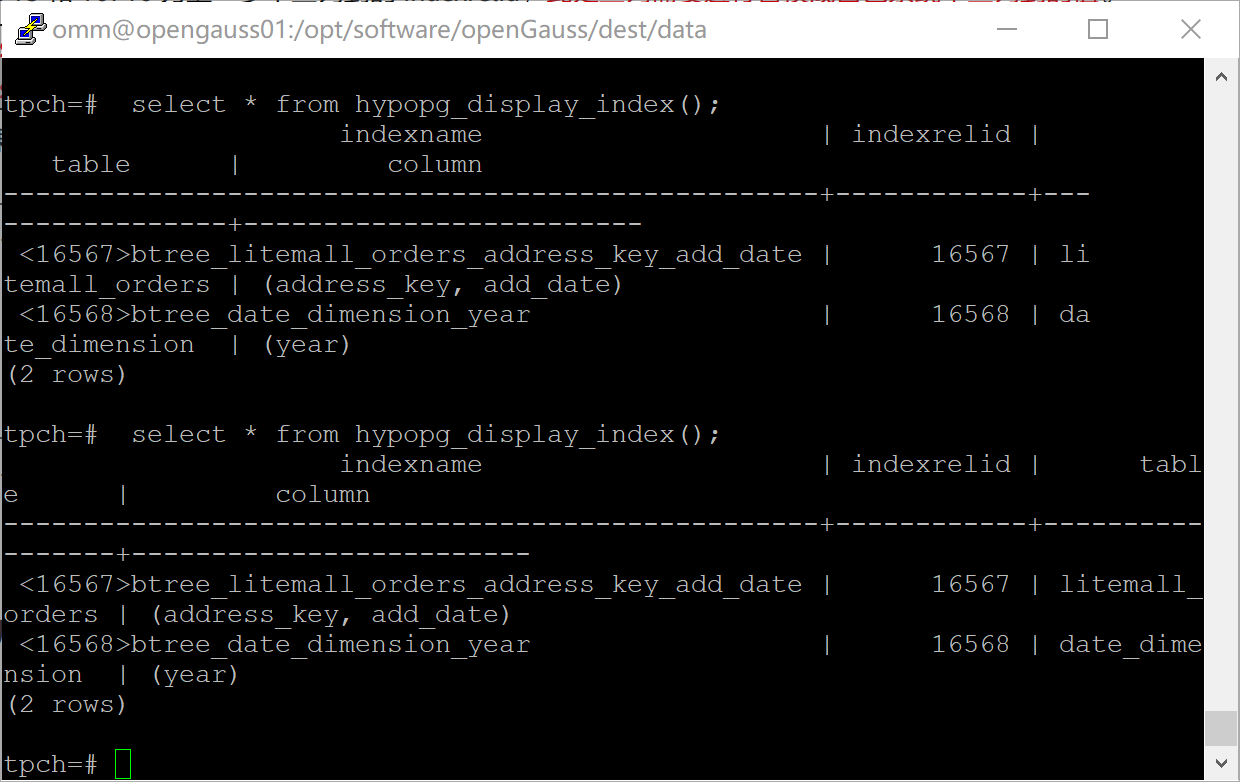
GROUP BY ad.province

ORDER BY SUM(o.actual\_price) DESC');



3. 查看创建的虚拟索引列，将执行结果截图。

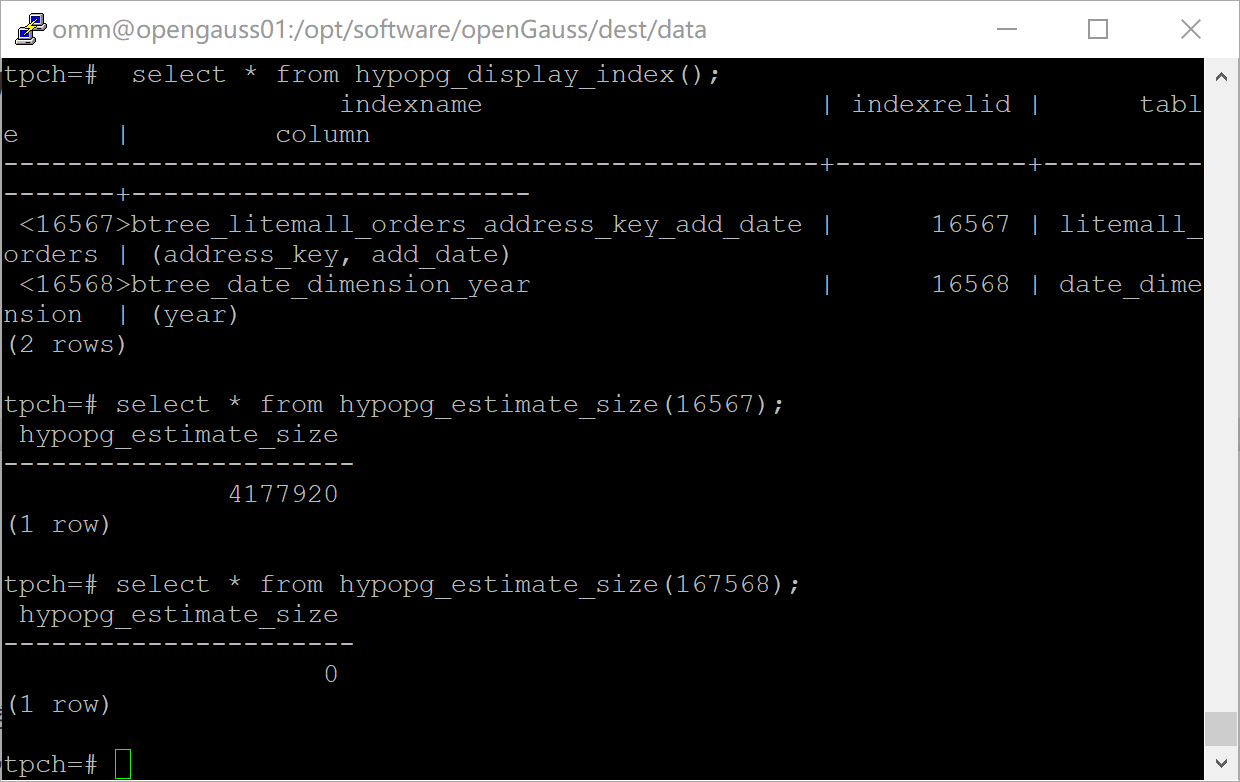
select \* from hypopg\_display\_index();



4. 获取索引虚拟列大小结果（单位为：字节），将执行结果截图。

select \* from hypopg\_estimate\_size(16715);

select \* from hypopg\_estimate\_size(16716);



5.再次使用explain，对该SQL加以分析，将执行结果截图。

EXPLAIN

SELECT ad.province AS province, SUM(o.actual\_price) AS GMV

FROM litemall\_orders o,

address\_dimension ad,

date\_dimension dd

WHERE o.address\_key = ad.address\_key

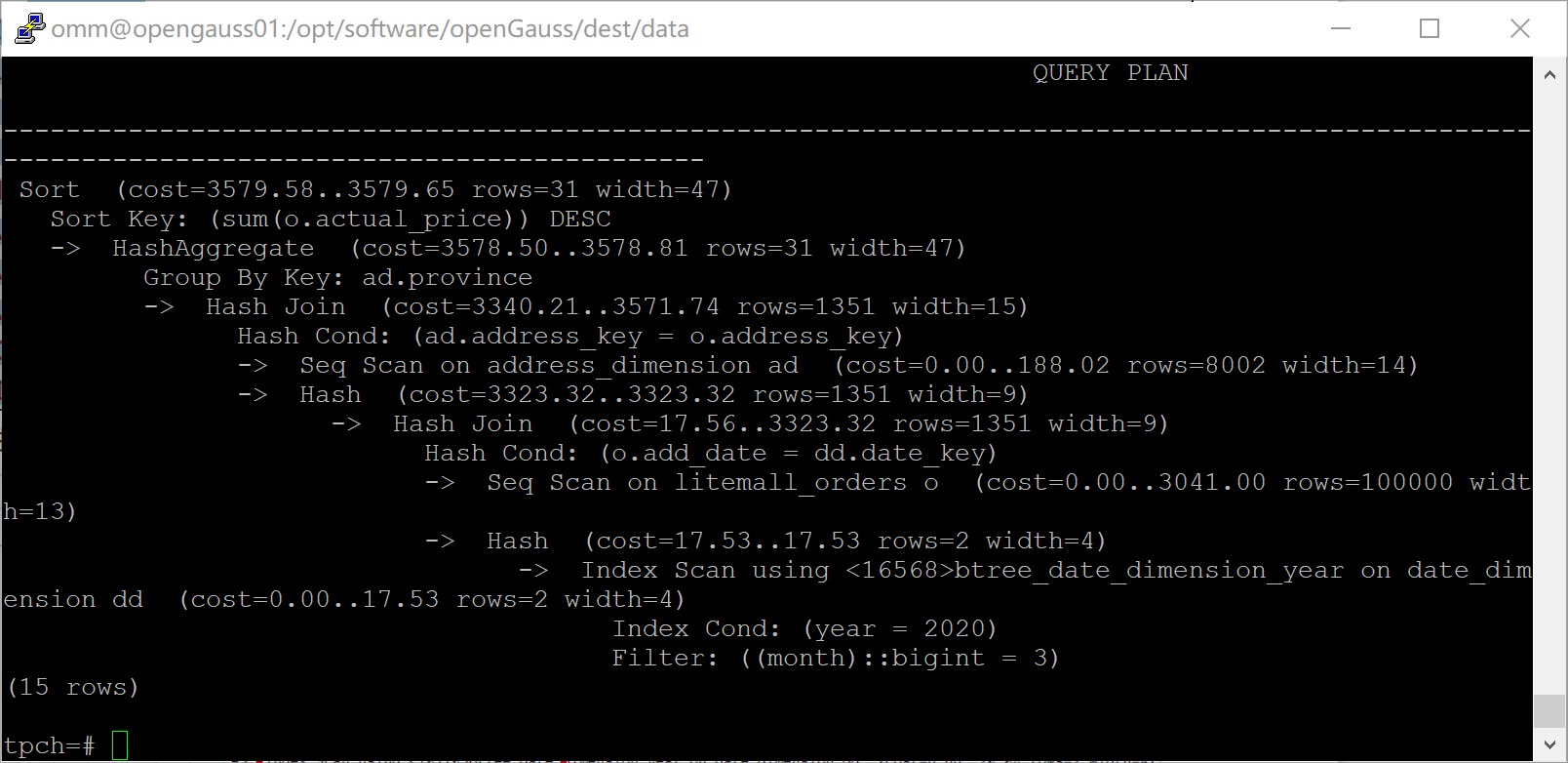
AND o.add\_date = dd.date\_key

AND dd.year = 2020

AND dd.month = 3

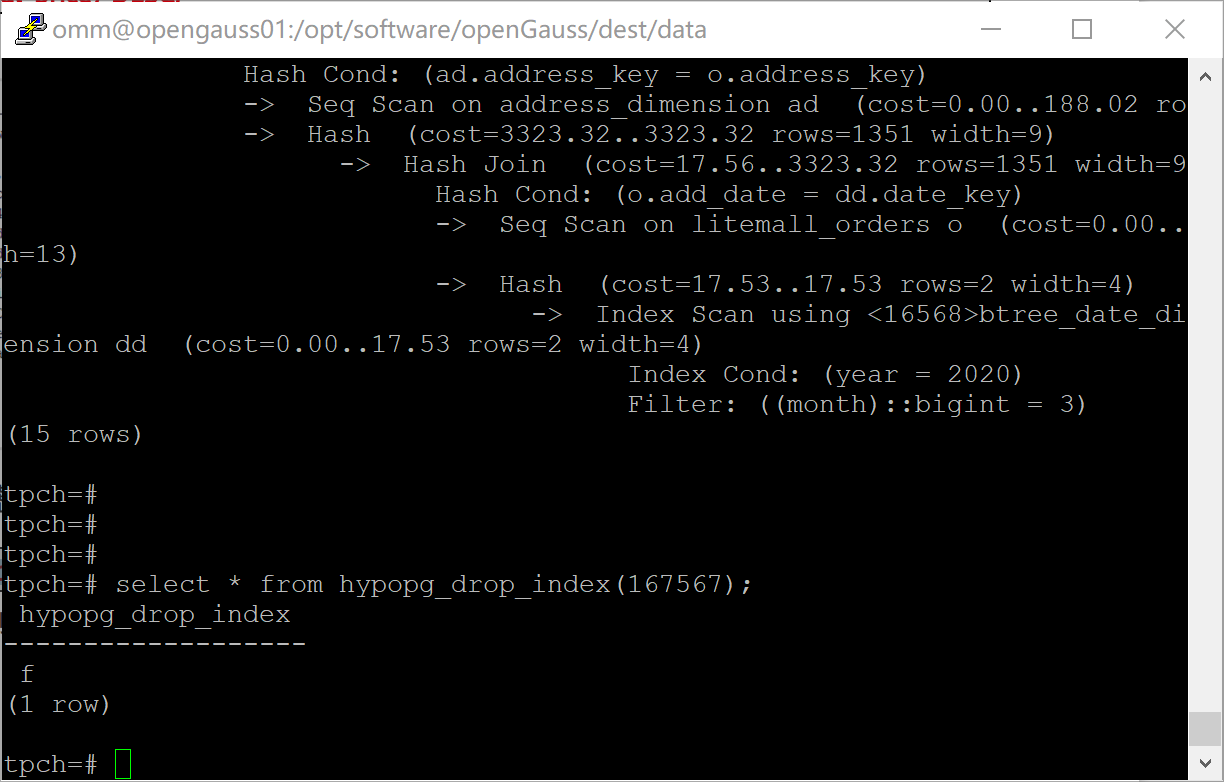
GROUP BY ad.province

ORDER BY SUM(o.actual\_price) DESC;



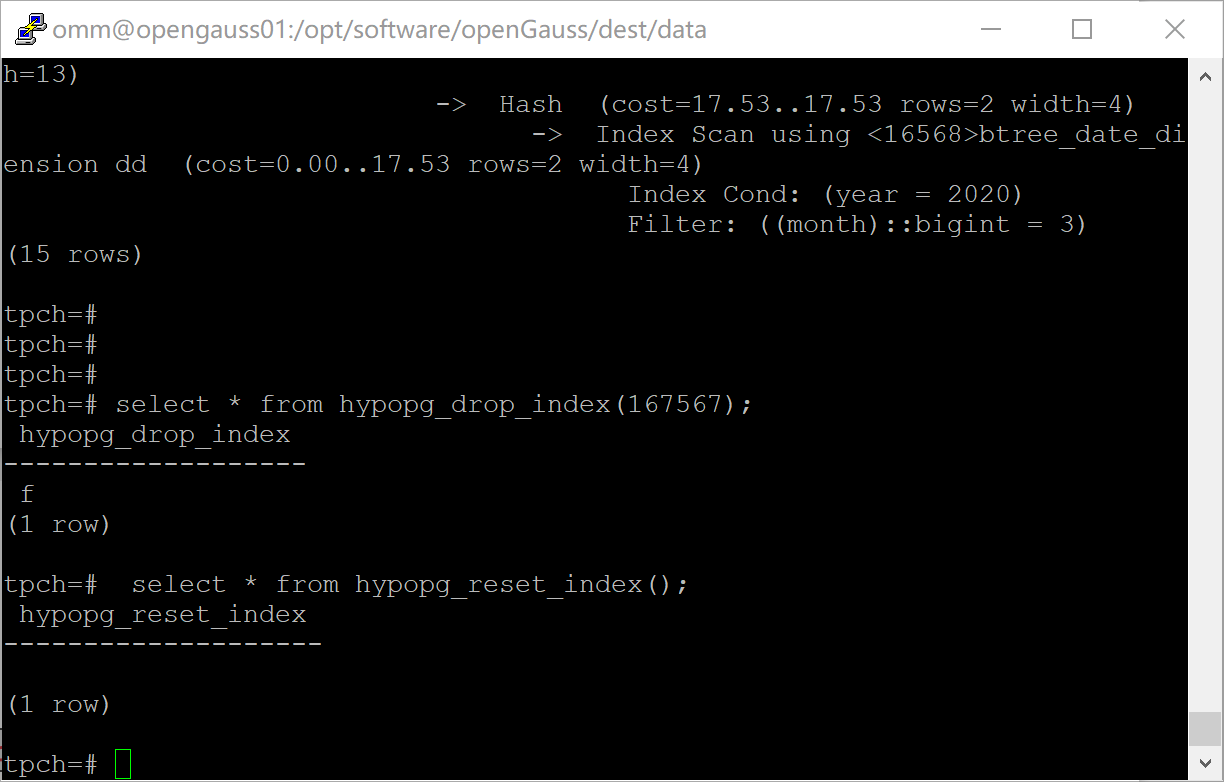
6. 删除某一个索引虚拟列，将执行结果截图。

select \* from hypopg\_drop\_index(16715);



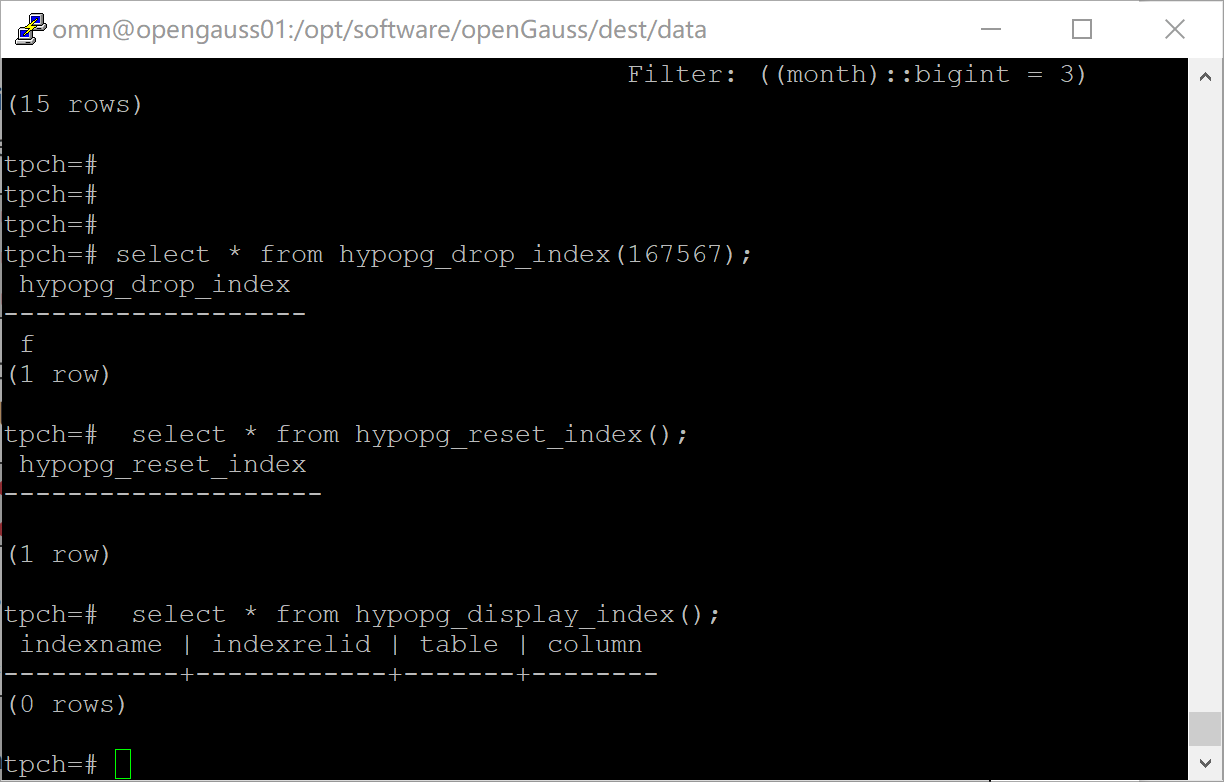
7. 删除某一个索引虚拟列，将执行结果截图。

select \* from hypopg\_reset\_index();



8. 查看索引虚拟列，将执行结果截图。

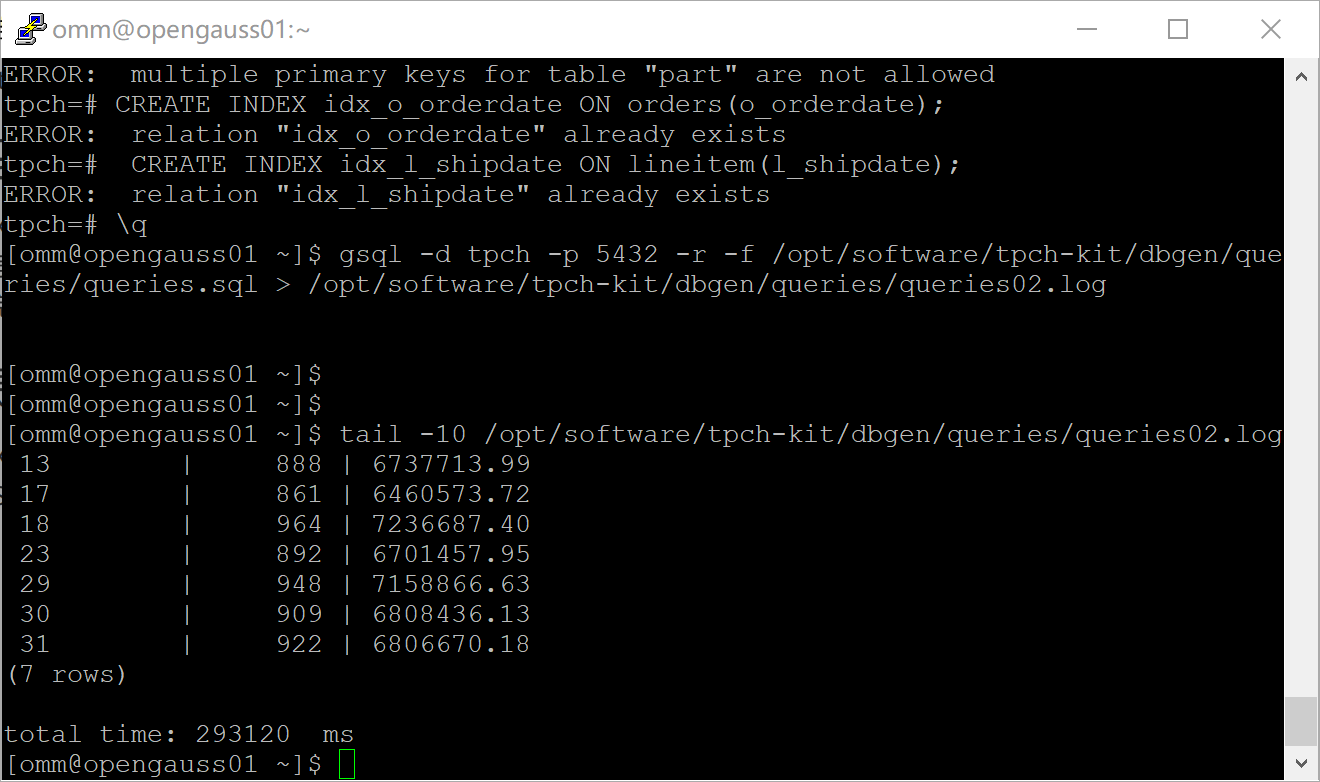
select \* from hypopg\_display\_index();



任务三：通过创建索引，对queries.sql中的SQL语句进行优化，并对比优化前后queries.sql执行的时间。

1. 重新执行queries.sql查询，将执行结果截图：

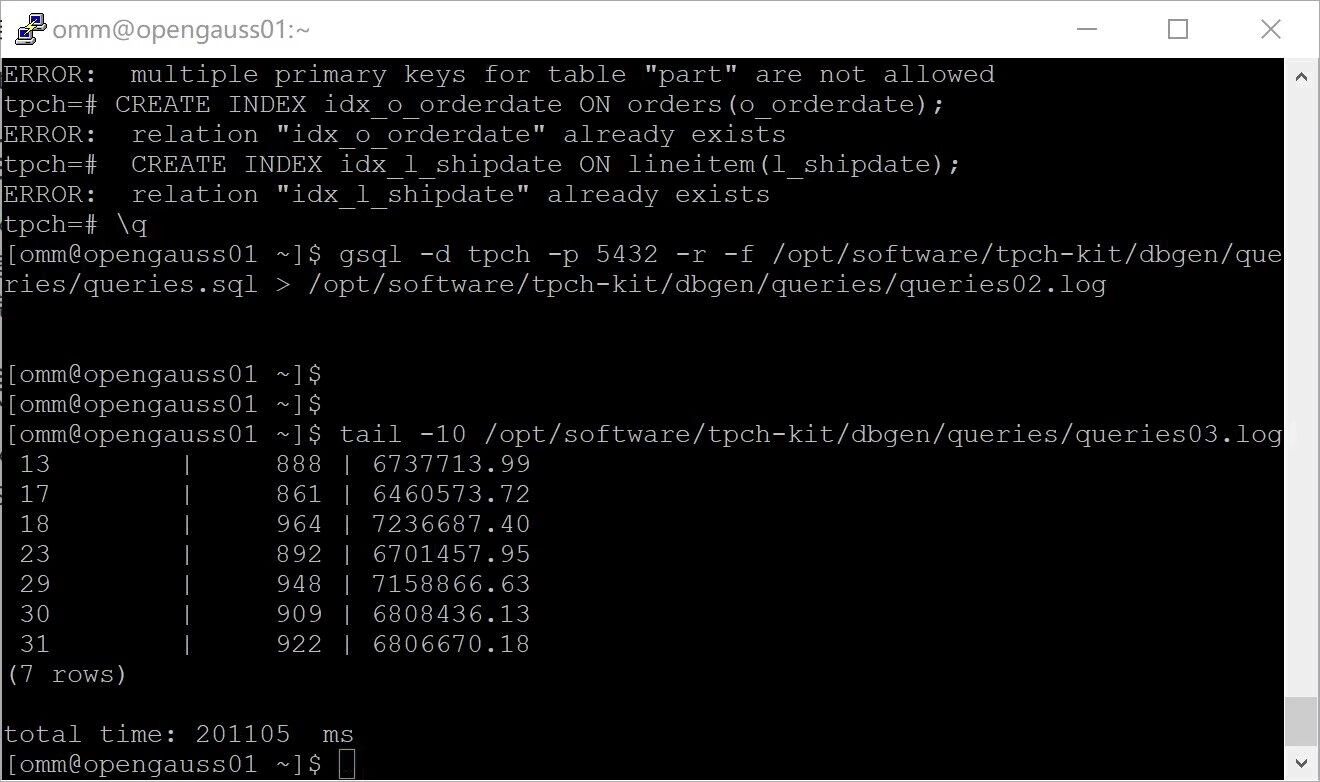
gsql -d tpch -p 5432 -r -f /opt/software/tpch-kit/dbgen/queries/queries.sql > /opt/software/tpch-kit/dbgen/queries/queries02.log



挑战一：进一步优化queries.sql中的查询语句，使得前后执行时间出现倍数级的提升。

1. 重新执行queries.sql查询，将执行结果截图：

gsql -d tpch -p 5432 -r -f /opt/software/tpch-kit/dbgen/queries/queries.sql > /opt/software/tpch-kit/dbgen/queries/queries03.log



实践思考题1：根据X-Tuner给出的参数优化，在哪些参数上进行了优化，为何要对这些参数进行优化？

答：（1）shared\_buffers：通过增加 shared\_buffers 参数的值，可以提高数据库系统对常用数据块的缓存能力，从而加快查询和数据访问的速度。

（2）max\_connections：通过适当调整 max\_connections 参数的值，可以根据系统的并发需求，平衡数据库连接的资源消耗和系统性能，避免过多的连接对系统造成压力。

（3）effective\_cache\_size：通过适当设置 effective\_cache\_size 参数的值，可以让优化器基于更准确的缓存估算值做出更优化的查询计划选择，提高查询的性能和响应时间。

（4）effective\_io\_concurrency：通过适当设置 effective\_io\_concurrency 参数的值，可以使数据库系统在进行磁盘I/O操作时实现更好的并行度，提高数据的读取和写入性能。

（5）wal\_buffers：通过增加 wal\_buffers 参数的值，可以提高写入WAL日志的效率，减少频繁的磁盘写入操作，提高数据库的写入性能和可靠性。

（6）random\_page\_cost：通过调整 random\_page\_cost 参数的值，可以使查询优化器更准确地估计随机I/O的成本，从而更准确地选择执行计划，提高查询的性能和效率。

（7）default\_statistics\_target：通过调整 default\_statistics\_target 参数的值，可以使数据库生成更准确的统计信息，用于查询优化和索引选择，从而提高查询性能和准确性。

实践思考题2：索引的使用，对于执行SQL有什么好处？除了使用索引和参数外，还有哪些方面可以对数据库进行优化？

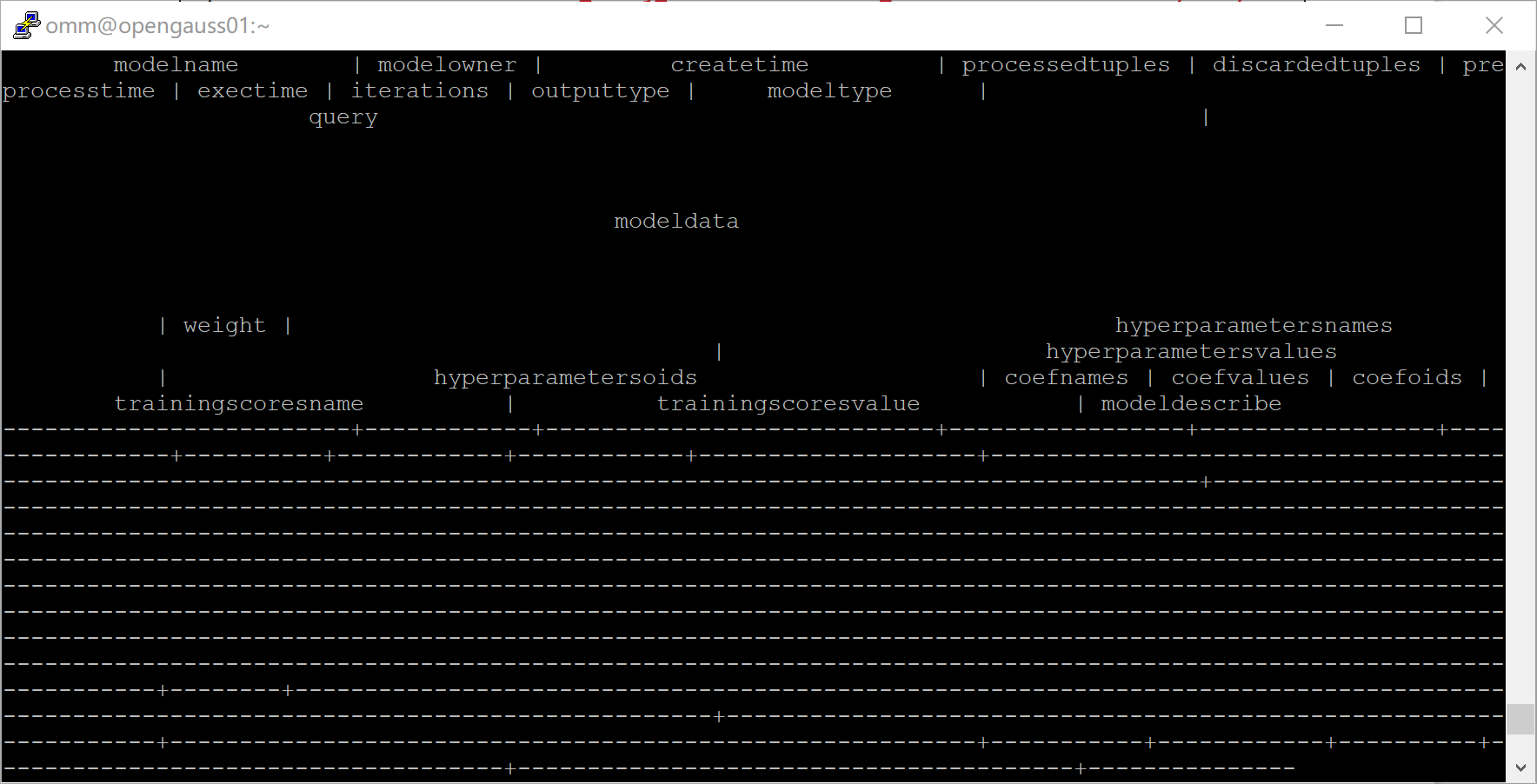
答：（1）好处：提高查询性能：索引可以加速数据的查找和检索，减少查询所需的读取操作；减少磁盘I/O操作；优化排序和连接操作：索引可以加速排序和连接操作。

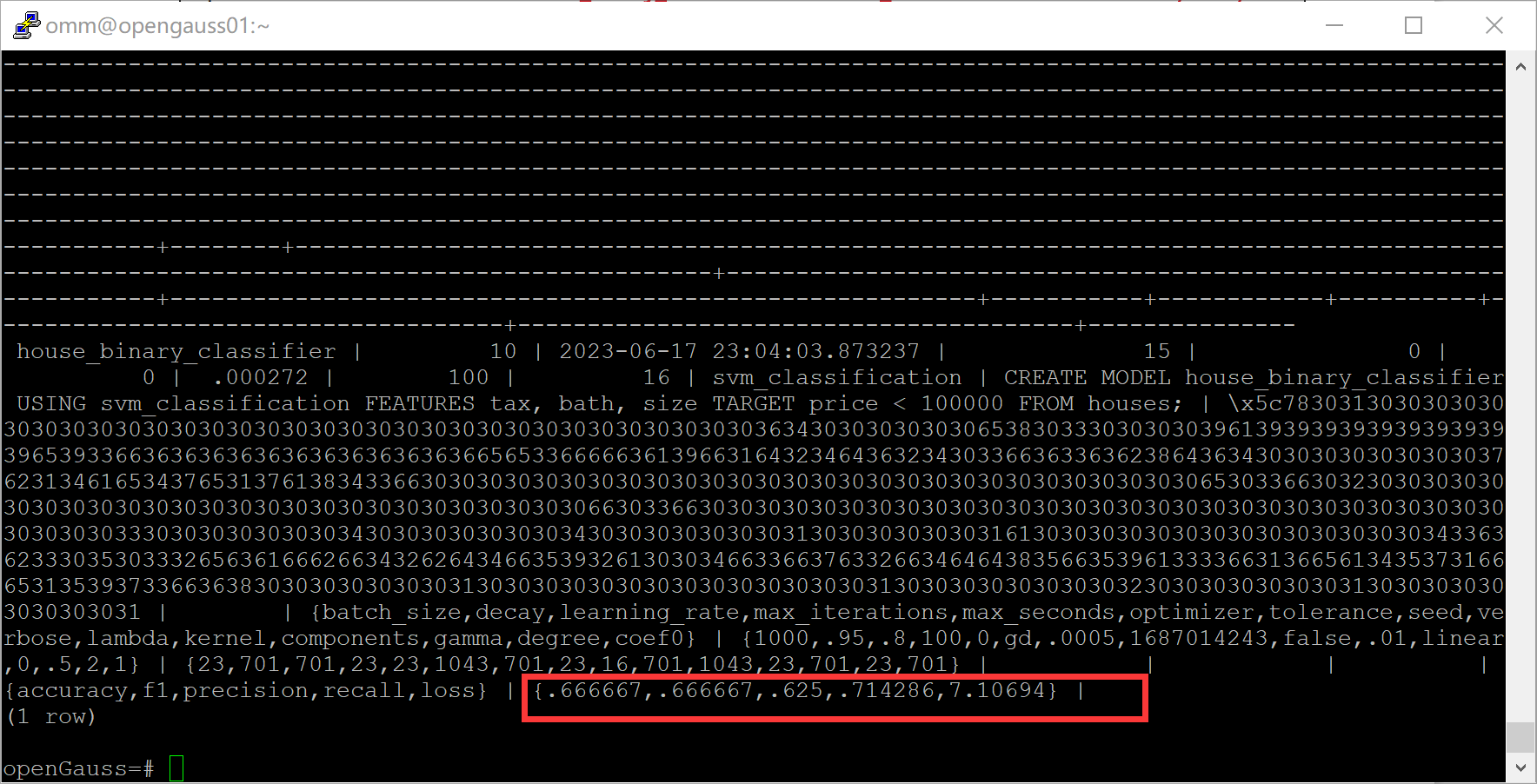
（2）数据库架构和设计优化：良好的数据库架构和设计可以提高数据库的性能和扩展性；数据库统计信息和查询优化：定期更新和维护数据库的统计信息，以提供准确的查询计划和索引选择；缓存和内存管理优化：合理配置数据库的缓存区域和内存分配，以减少磁盘I/O操作，提高数据的读取和写入性能。

# 关卡四、openGauss的DB4AI特性应用

任务一：在gs\_model\_warehouse系统表中查看训练后的模型信息，将执行结果截图：

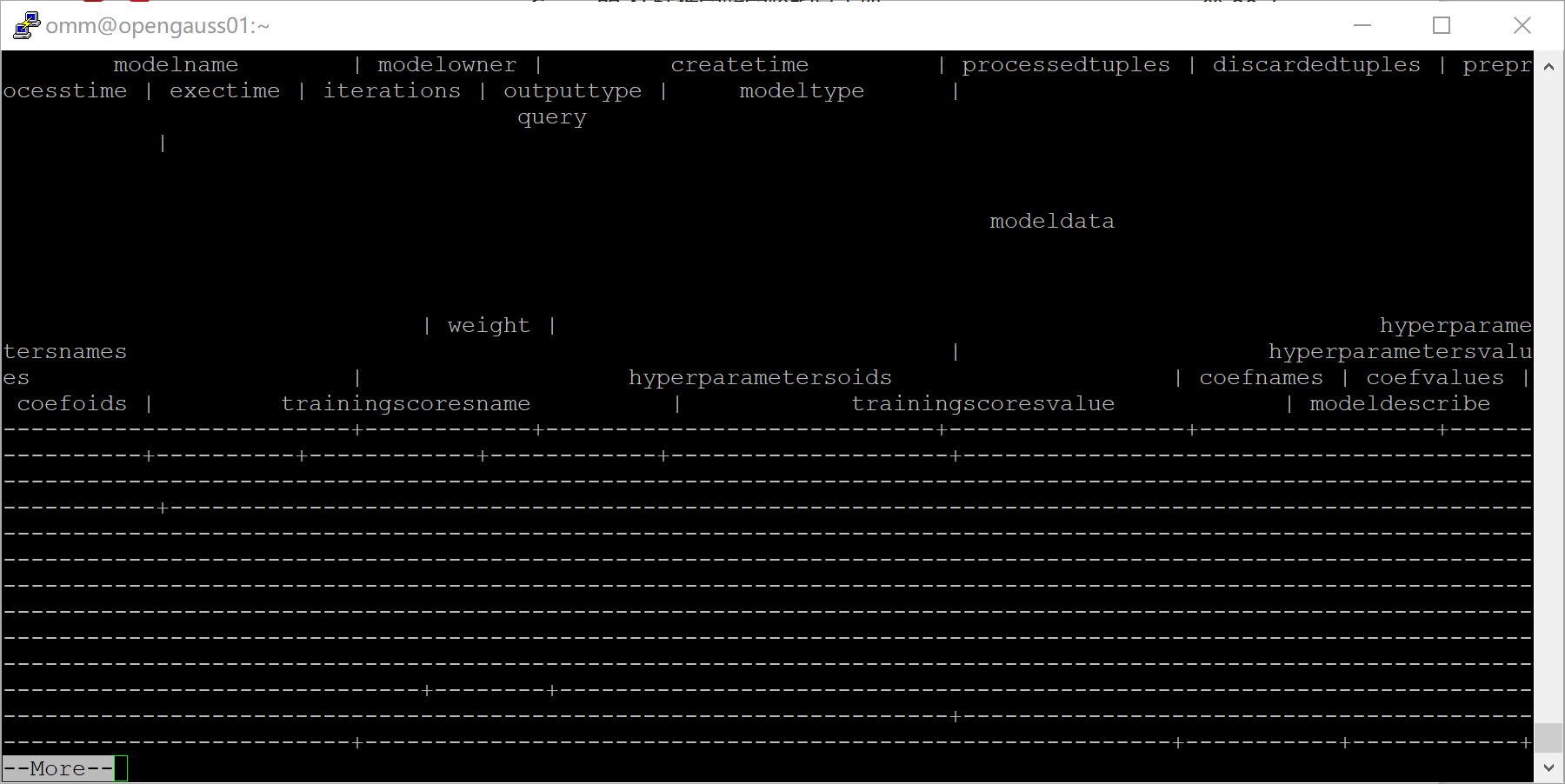
postgres=# SELECT \* FROM gs\_model\_warehouse WHERE modelname = 'house\_binary\_classifier';

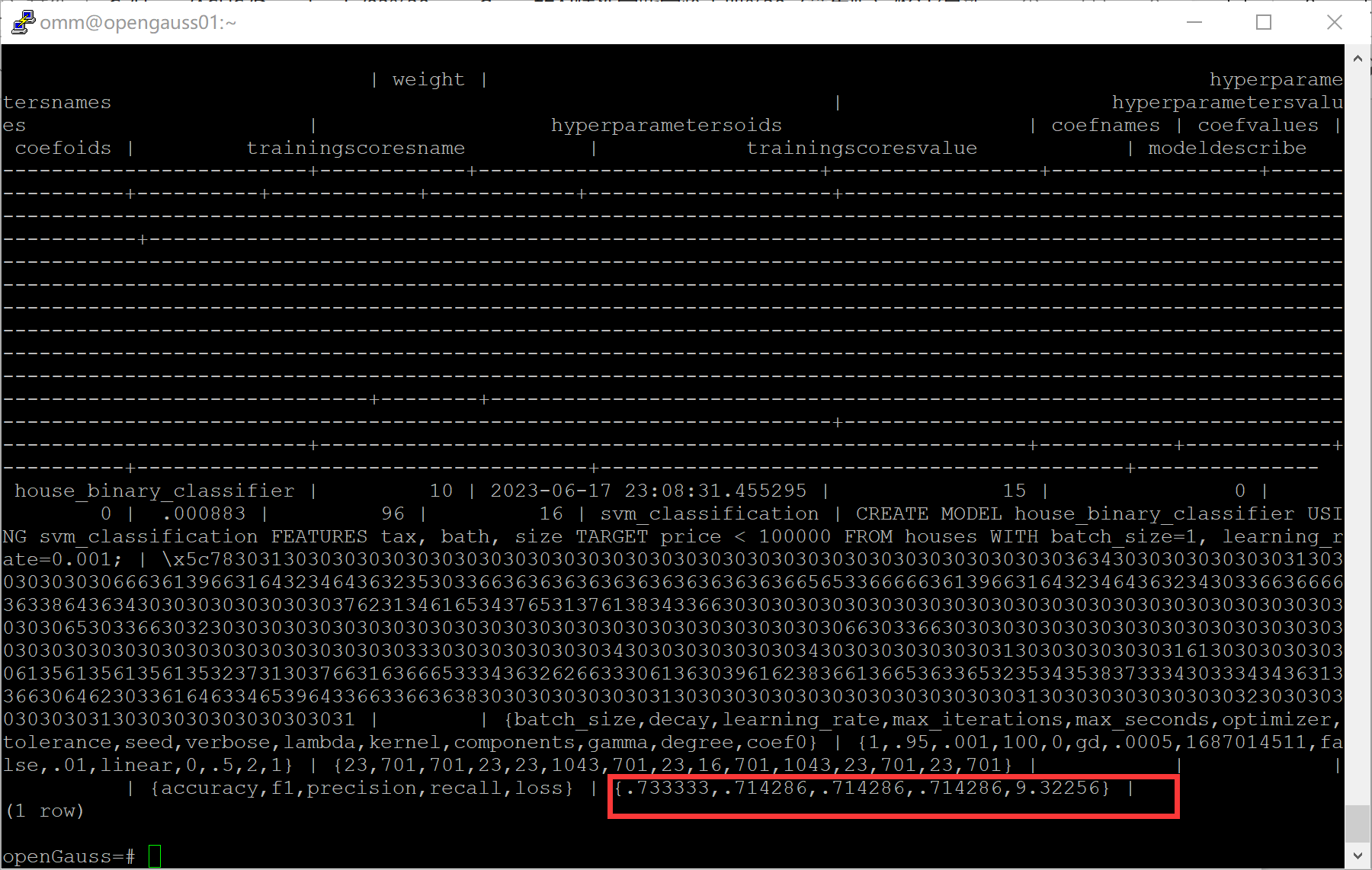




任务二：观察新模型的信息，将执行结果截图。

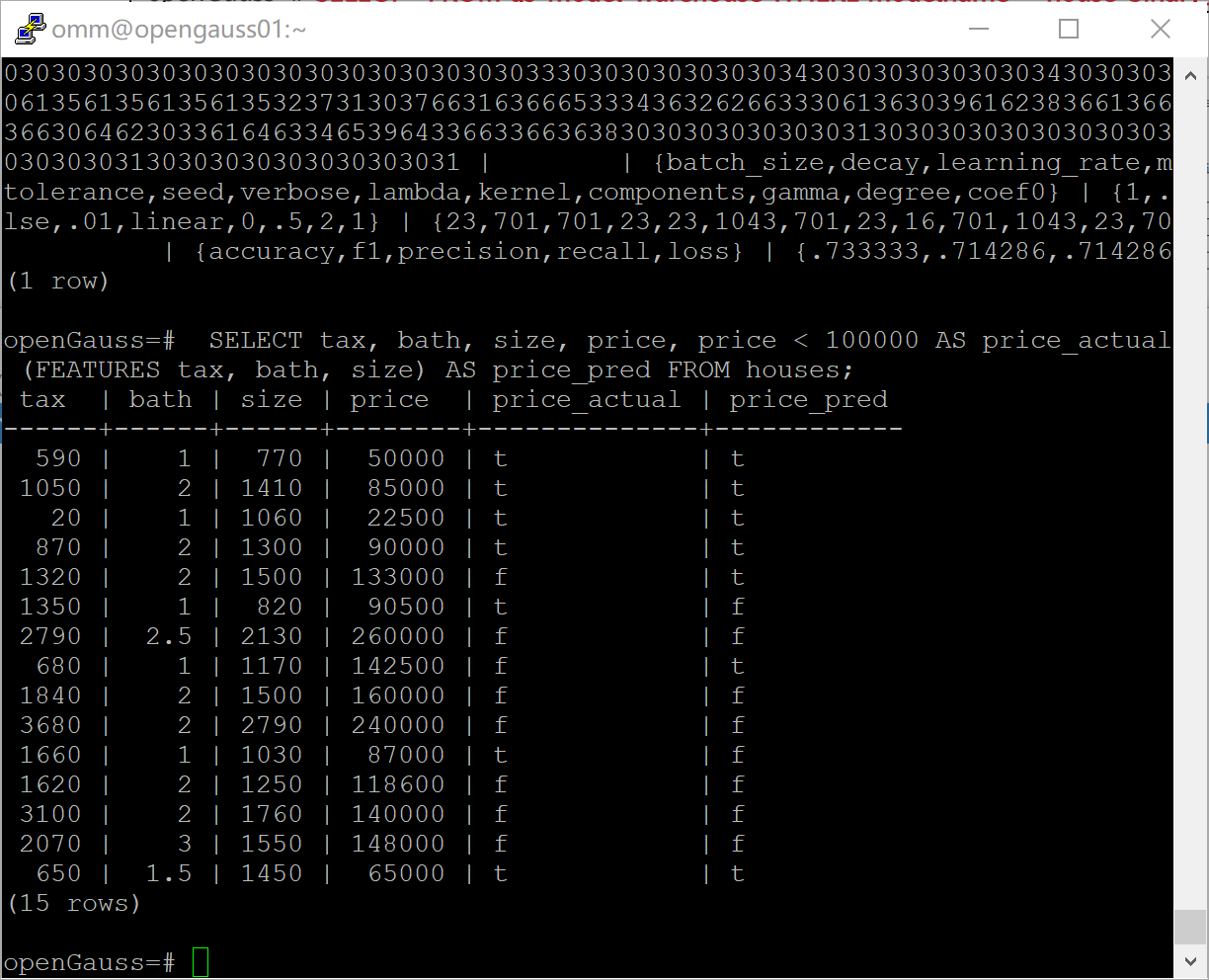
postgres=# SELECT \* FROM gs\_model\_warehouse WHERE modelname = 'house\_binary\_classifier';





任务三：利用训练好的逻辑回归模型预测数据，并与SVM算法进行比较，将执行结果截图。

postgres=# SELECT tax, bath, size, price, price < 100000 AS price\_actual, PREDICT BY house\_binary\_classifier (FEATURES tax, bath, size) AS price\_svm\_pred, PREDICT BY house\_logistic\_classifier (FEATURES tax, bath, size) AS price\_logistic\_pred FROM houses;



实践思考题1：分类模型与回归模型有何不同？

答：（1）目标类型：分类模型用于预测离散的类别或标签；回归模型：用于预测连续的数值，回归模型的输出是对目标变量的数值预测。

（2）输出结果类型：分类模型输出是离散的类别标签或概率。分类模型可以将样本分为不同的类别，并为每个类别分配一个概率或置信度；回归模型输出是连续的数值。回归模型预测的是目标变量的实际数值。

（3）模型选择和评估指标：分类模型：通常使用准确率、精确率、召回率、F1得分等指标来评估分类模型的性能。回归模型：常用的评估指标包括均方误差（MSE）、均方根误差（RMSE）、平均绝对误差（MAE）等。

实践思考题2：什么是SVM算法？

答：SVM是一种机器学习算法，常用于进行分类和回归分析。SVM 的主要目标是找到一个最优的超平面（或超曲面），将数据样本在高维空间中划分为不同的类别。SVM 的基本思想是通过寻找一个最优的决策边界，使得不同类别的样本尽可能地被最大间隔分开。这个决策边界被称为超平面。

实践思考题3：分类问题有哪些评价指标，请分别说明他们的含义？

答：（1）准确率（Accuracy）：准确率是最常见的评价指标，表示分类模型正确预测的样本数量与总样本数量之间的比例。

（2）精确率（Precision）：精确率衡量了模型在预测为正类的样本中有多少是真正的正类。计算公式为：精确率 = (真正的正类样本数) / (真正的正类样本数 + 假正类样本数)。

（3）召回率（Recall）：召回率衡量了模型能够正确识别出正类样本的能力。计算公式为： 召回率 = (真正的正类样本数) / (真正的正类样本数 + 假负类样本数)。

（4）F1得分（F1 Score）：F1得分是精确率和召回率的综合指标，综合考虑了分类模型的准确性和召回性能。计算公式为：F1得分 = 2 \* (精确率 \* 召回率) / (精确率 + 召回率)。

（5）ROC-AUC：ROC曲线是衡量分类模型性能的一种方法，AUC是 ROC 曲线下的面积。ROC 曲线以真正类率（TPR，召回率的另一个名称）为纵轴，假正类率（FPR，1-特异度）为横轴，可以通过调整分类阈值来获得不同的点。AUC 表示 ROC 曲线下的面积，取值范围在0到1之间，值越大表示模型分类性能越好。

实践思考题4：回归问题有哪些评价指标，请分别说明他们的含义？

答：（1）均方误差（MSE）：均方误差衡量了预测值与实际值之间的平均差异的平方。

（2）均方根误差（RMSE）：均方根误差是MSE的平方根，它衡量了预测值与实际值之间的平均差异的平方根。

（3）平均绝对误差（MAE）：平均绝对误差衡量了预测值与实际值之间的平均绝对差异。

MAE衡量了预测值与实际值之间的平均绝对差异的大小，它对异常值不敏感。

（4）R平方（R-squared）：R平方衡量了模型对目标变量的变异程度的解释能力，即拟合优度。它表示预测值与实际值之间的变异程度占总变异程度的比例。

（5）调整R平方（Adjusted R-squared）：调整R平方对模型复杂度进行了调整，考虑了自变量的数量和样本数量。它惩罚了增加自变量带来的模型复杂度。