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

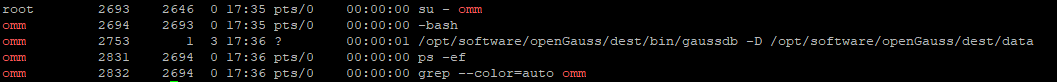
1. 查询数据库状态成功截图

屏幕上有字

描述已自动生成

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？

通过源码安装可以自定义配置

# 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;;

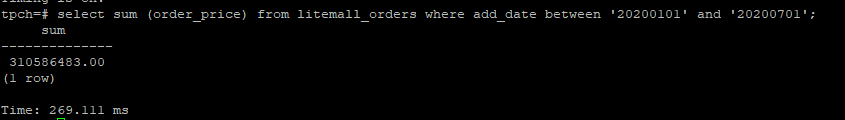
文本

描述已自动生成

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';

文本

描述已自动生成

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';

文本

描述已自动生成

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

文本

描述已自动生成

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;

文本

描述已自动生成

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

文本

描述已自动生成

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;

文本

描述已自动生成

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;

图片包含 图形用户界面

描述已自动生成

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;

文本

描述已自动生成

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;

文本

描述已自动生成

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;

图片包含 图示

描述已自动生成

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;

图片包含 图示

描述已自动生成图片包含 图示

描述已自动生成

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？

行存表将数据按照行的方式存储，每一行包含多个列。当执行需要获取整行数据的SQL语句时，由于行存表的数据读取是按照行的方式进行，因此行存表的执行效率更高。

而列存表将数据按照列的方式存储，每一列包含多个行。当执行需要获取多个列中的数据的SQL语句时，由于列存表的数据读取是按照列的方式进行，因此列存表的执行效率更高。

在执行查询大量数据的SQL语句时，列存表效率更高，因为列存表可以只读取需要查询的列，而无需读取整行数据。在执行聚合函数操作时，列存表也比行存表效率更高，因为列存表可以只读取需要进行聚合操作的列，而无需读取整行数据。在执行插入、更新和删除操作时，行存表效率更高，因为这些操作需要涉及到整行的数据。

实践思考题2：全量物化视图与增量物化视图有哪些差别？

全量物化视图适用于数据量较小但查询频率高的场景，而增量物化视图则适用于数据量较大且更新频率较高的场景。全量物化视图需要占用较大的存储空间，但查询速度较快；增量物化视图则需要对增量数据进行跟踪和维护，但可以减少存储空间，并且能够保证数据的实时性。

# 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

文本

描述已自动生成

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

电脑屏幕的照片上有文字

中度可信度描述已自动生成

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'

文本

描述已自动生成

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;

文本

描述已自动生成

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

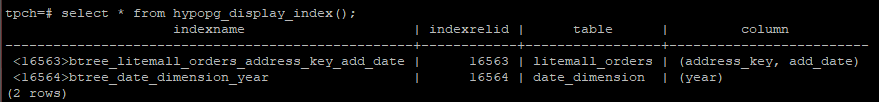
ORDER BY SUM(o.actual\_price) DESC');

日程表

描述已自动生成

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);

文本

描述已自动生成

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;

文本

描述已自动生成

6. 删除某一个索引虚拟列，将执行结果截图。

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

文本

描述已自动生成

7. 删除某一个索引虚拟列，将执行结果截图。

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

文本

描述已自动生成

8. 查看索引虚拟列，将执行结果截图。

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

日程表

低可信度描述已自动生成

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

屏幕上写着字

描述已自动生成

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？

对shared\_buffers、max\_connections、effective\_cache\_size、effective\_io\_concurrency、wal\_buffers、random\_page\_cost和default\_statistics\_target进行了参数优化，通过优化可提高查询语句运行的效率。

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？

使用索引可以提高SQL查询的执行速度，因为索引可以快速定位到符合查询条件的数据，而不需要全表扫描。使用索引还可以减少数据库的I/O操作，因为索引可以存储在内存中，而数据表则需要从磁盘中读取，所以使用索引可以减少读取磁盘的次数。除了索引和参数外，还可以优化数据库设计和储存方式。

# 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';



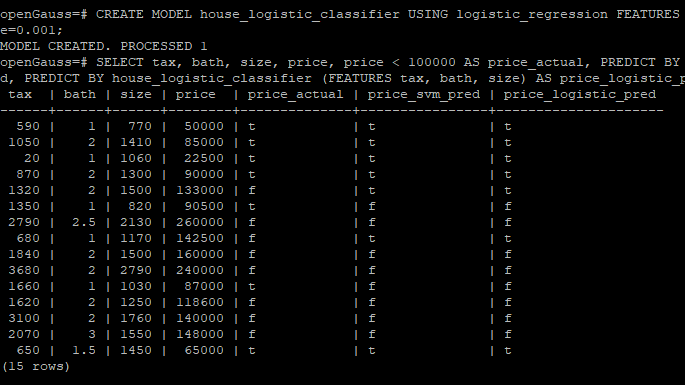
任务二：观察新模型的信息，将执行结果截图。

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？

分类模型和回归模型是两种不同的机器学习模型。分类模型是用于分类任务的模型，其目标是预测输入数据属于哪个类别或标签。而回归模型则是用于预测连续数值型数据的模型，其目标是预测输出值的数值。

实践思考题2：什么是SVM算法？

在分类问题中，SVM算法的目标是找到一个超平面，将特征空间划分为不同的类别，使得不同类别之间的间隔尽可能大；在回归问题中，SVM算法的目标是找到一个超平面，使得这个超平面与样本的距离最小。

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？

1、准确率：准确率是分类模型最常用的评价指标，它计算预测正确的样本数除以总样本数。简单来说，准确率指的是分类器正确分类的样本在总样本数中所占的比例。准确率越高，分类器的性能越好。但当正负样本不平衡时，准确率不能很好地反映分类器的性能。

2、精确率：精确率是指被分类器正确预测为正例的样本在所有预测为正例的样本中所占的比例。 精确率越高，分类器将负样本分为正样本的能力越小。

3、召回率：召回率是指被分类器正确预测为正例的样本在所有真实为正例的样本中所占的比例。召回率越高，分类器能够正确找到正样本的数量越多。

4、F1值：F1值是精确率和召回率的调和平均数。它综合考虑了精确率和召回率的贡献，是一个综合评价指标。F1值越高，分类器的性能越好。

5、ROC曲线与AUC值：ROC曲线是二分类问题中用于绘制分类器性能的曲线，横坐标为假正率，即被错误地分类为正例的负例占所有负例的比例。纵坐标为真正率，即被正确分类为正例的正例占所有正例的比例。AUC值则是ROC曲线下的面积，它可以反映分类器的分类能力，AUC值越大，分类器的性能越好。

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？

1、均方误差：是真实值与预测值之差的平均值的平方，常用于连续型变量。MSE越小，说明预测结果越准确。

2、均方根误差：是MSE的平方根，可以直观地反映出预测值与真实值之间的偏差大小。RMSE越小，说明预测结果越准确。

3、平均绝对误差：是真实值与预测值之差的绝对值的平均值，也常用于连续型变量。MAE越小，说明预测结果越准确。

4、决定系数：是预测值与真实值协方差的平方与预测值方差的乘积的商，表示回归模型对变量变化的解释能力。R²越接近1，说明模型对数据的拟合能力越强。

5、平均绝对百分比误差：是预测值与真实值之间的百分比误差的平均值，常用于对回归模型预测质量的评估。MAPE越小，说明预测结果越准确。

6、对数均方误差：是真实值与预测值的对数之差的平方的均值，适用于预测值和真实值的范围都较大的情况。MSLE越小，说明预测结果越准确。