openGauss AI特性创新实践课



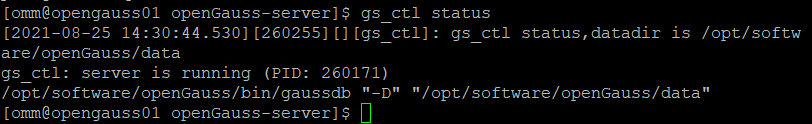
华为技术有限公司

# 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：

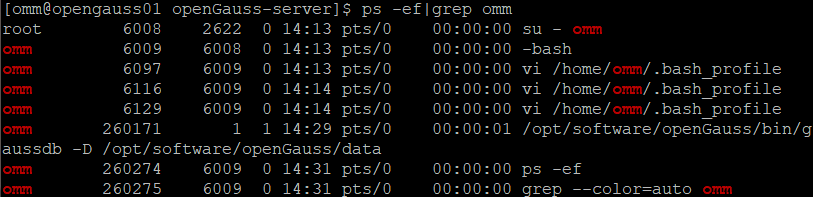
任务一：数据库状态验证

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？

1、可以安装指定的版本

2、可以指定编译参数

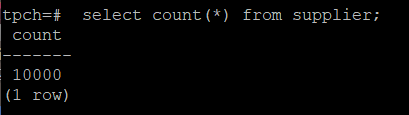
3、不会对系统造成污染

# 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：

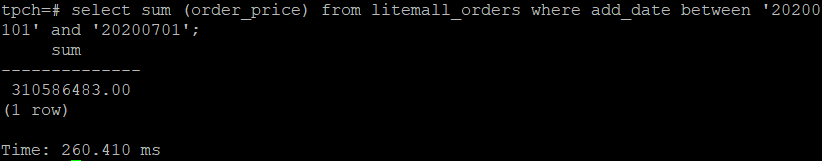
select count(\*) from supplier;;



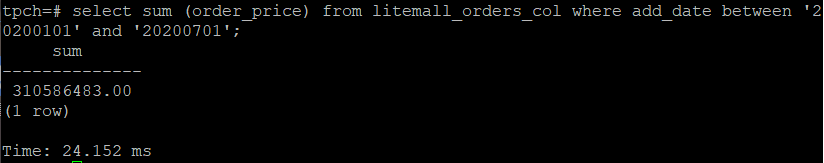
任务二：行存表与列存表执行效率对比

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';

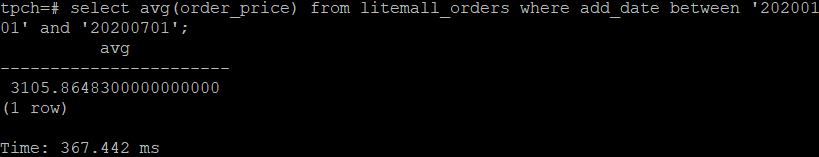


select sum (order\_price) from litemall\_orders\_col where add\_date between '20200101' and '20200701';

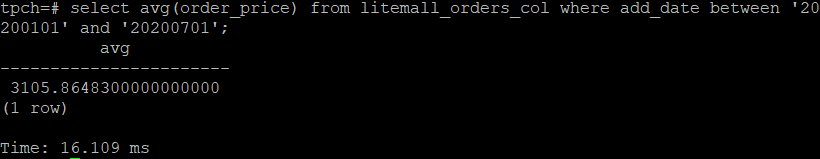


2. 2020年上半年litemall\_orders行存表与litemall\_orders\_col列存表中的order\_price的平均值查询，并对比执行效率截图

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

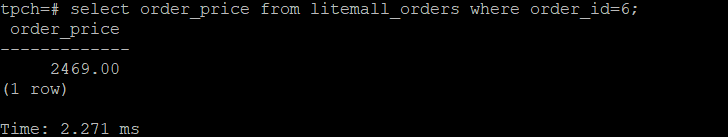


select avg (order\_price) from litemall\_orders\_col where add\_date between '20200101' and '20200701';

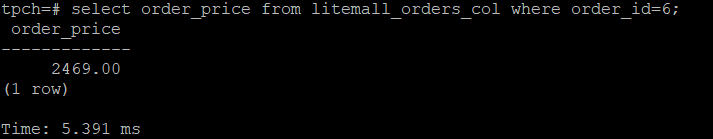


3. 查询litemall\_orders行存表与litemall\_orders\_col列存表中order\_id为6的order\_price的值，并对比执行效率截图。

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;

C:\Users\csq\AppData\Local\Temp\WeChat Files\30b2fbe572ada0d22732fe7a50afe15.png

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

C:\Users\csq\AppData\Local\Temp\WeChat Files\81604c51baf4a46d44e5ffa10fa1eaa.png

任务三：物化视图的使用

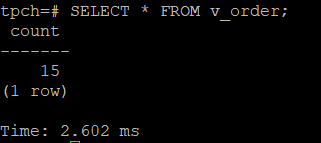
1. 创建物化视图所需要的表后，对表内容进行查询，对查询结果截图：

SELECT \* FROM test\_view;



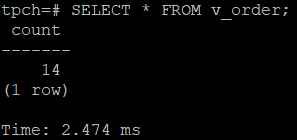
2. 使用物化视图统计人数，查询物化视图结果，将执行结果截图。

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？

行存表需要频繁插入或更新的操作，其操作与索引和行的大小更为相关；列存表可在数据列中高效查找数据，无需维护索引(任何列都能作为索引)，查询过程中能够尽量减少无关IO，避免全表扫描。

基于一列或比较少的列计算的时候；经常关注一张表某几列而非整表数据的时候；

数据表拥有非常多的列的时候；数据表有非常多行数据并且需要聚集运算的时候；数据表列里有非常多的重复数据，有利于高度压缩。

实践思考题2：全量物化视图与增量物化视图有哪些差别？

全量物化视图仅支持对创建好的物化视图做全量更新，而不支持做增量更新；

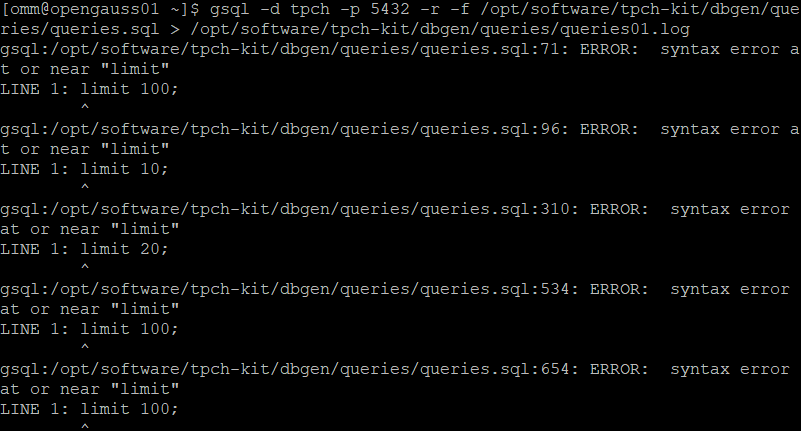
增量物化视图顾名思义就是可以对物化视图增量刷新，需要用户手动执行语句完成对物化视图在一段时间内的增量数据进行刷新。与全量创建物化视图不同在于目前增量物化视图所支持场景较小，目前物化视图创建语句仅支持基表扫描语句或者UNION ALL语句。

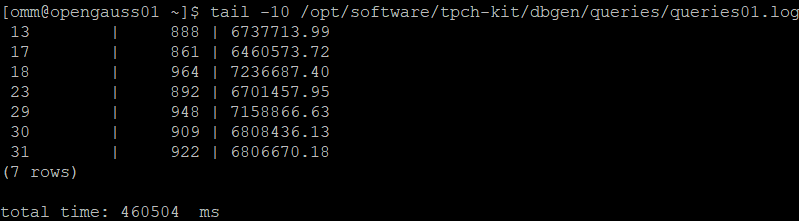
# 关卡三、openGauss的AI4DB特性应用

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：

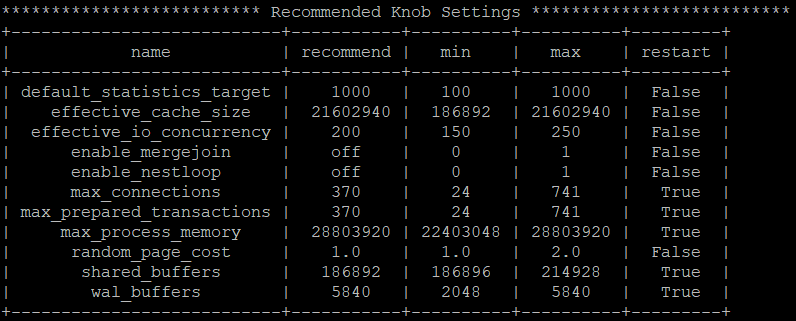
gsql -d tpch -p 5432 -r -f /opt/software/tpch-kit/dbgen/queries/queries.sql > /opt/software/tpch-kit/dbgen/queries/queries01.log





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

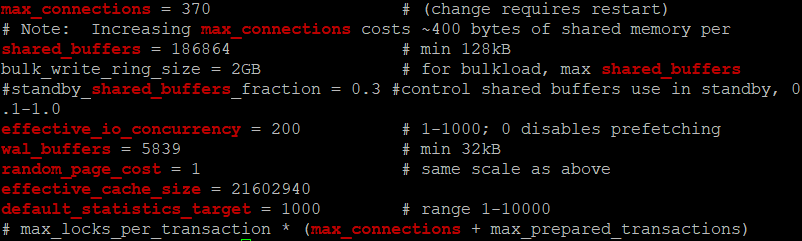
gs\_xtuner recommend --db-name tpch --db-user omm --port 5432 --host 127.0.0.1 --host-user omm



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

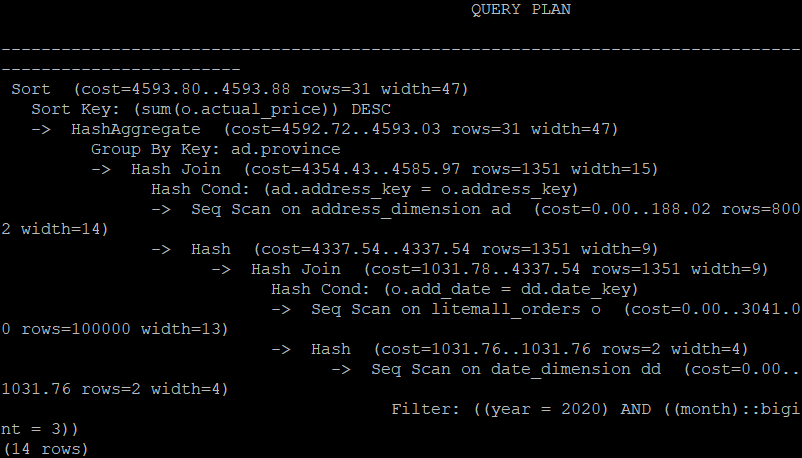
AND o.add\_date = dd.date\_key

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

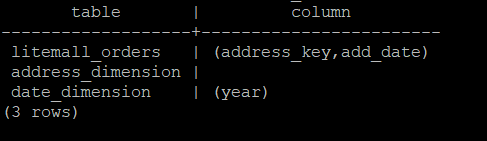
AND o.add\_date = dd.date\_key

AND dd.year = 2020

AND dd.month = 3

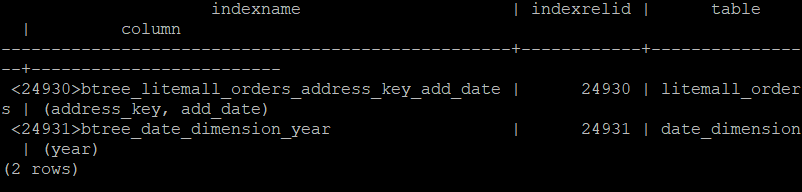
GROUP BY ad.province

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。

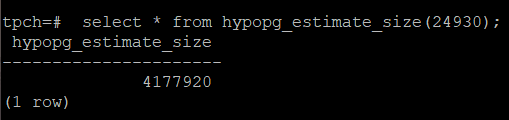
select \* from hypopg\_display\_index();

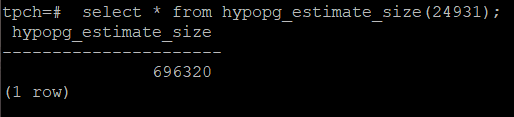


4. 获取索引虚拟列大小结果（单位为：字节），将执行结果截图。

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

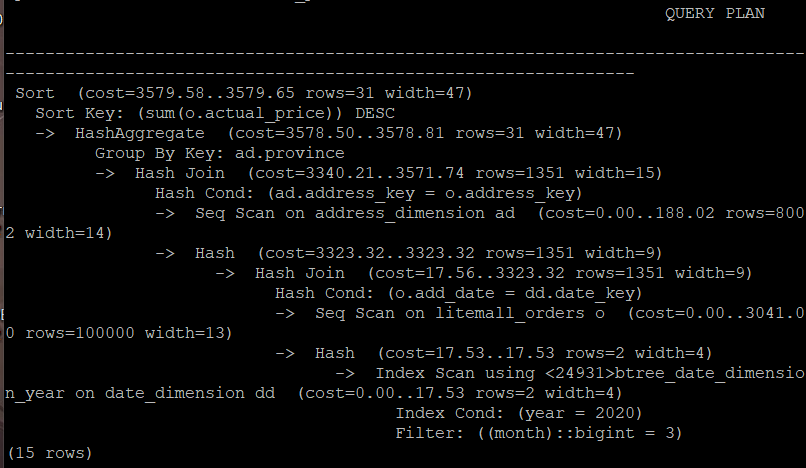
AND o.add\_date = dd.date\_key

AND dd.year = 2020

AND dd.month = 3

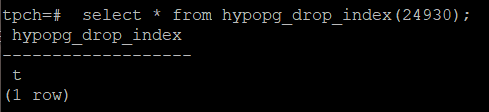
GROUP BY ad.province

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



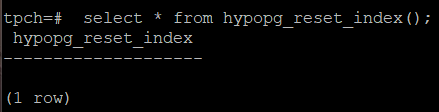
6. 删除某一个索引虚拟列，将执行结果截图。

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



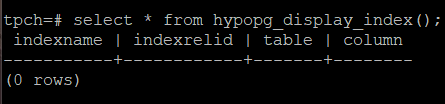
7. 删除某一个索引虚拟列，将执行结果截图。

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



8. 查看索引虚拟列，将执行结果截图。

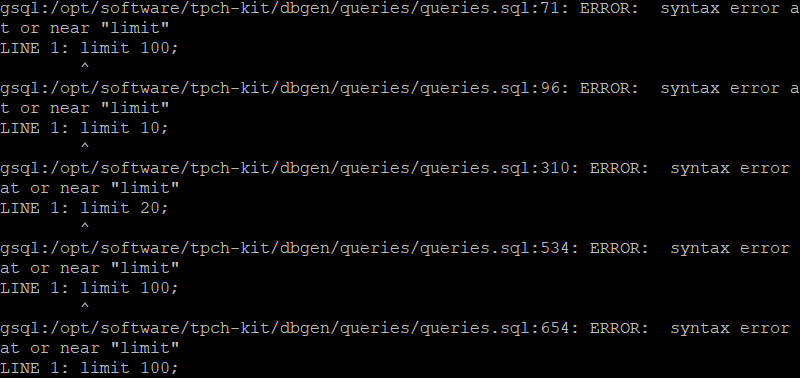
select \* from hypopg\_display\_index();

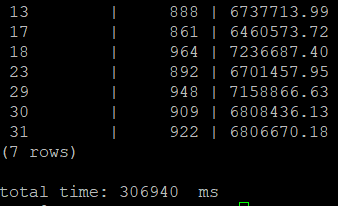


任务三：通过创建索引，对queries.sql中的SQL语句进行优化，并对比优化前后queries.sql执行的时间。

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？

在shared\_buffers, max\_connections, effective\_cache\_size, effective\_io\_concurrency, wal\_buffers, random\_page\_cost, default\_statistics\_target上进行了优化。

1、shared\_buffers

PostgreSQL既使用自身的缓冲区，也使用内核缓冲IO。这意味着数据会在内存中存储两次，首先是存入PostgreSQL缓冲区，然后是内核缓冲区。这被称为双重缓冲区处理。对大多数操作系统来说，这个参数是最有效的用于调优的参数。

2、wal\_buffers

PostgreSQL将其WAL（预写日志）记录写入缓冲区，然后将这些缓冲区刷新到磁盘。由wal\_buffers定义的缓冲区的默认大小为16MB，但如果有大量并发连接的话，则设置为一个较高的值可以提供更好的性能。

3、effective\_cache\_size

effective\_cache\_size提供可用于磁盘高速缓存的内存量的估计值。它只是一个建议值，而不是确切分配的内存或缓存大小。它不会实际分配内存，而是会告知优化器内核中可用的缓存量。在一个索引的代价估计中，更高的数值会使得索引扫描更可能被使用，更低的数值会使得顺序扫描更可能被使用。

4、random\_page\_cost

此设置规定，非连续牵强磁盘页面将有成本，而且直接影响到查询规划决策。使用高等待时间的存储时，如旋转磁盘用保守的值会是特别重要的。

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？

1、大大加快数据的检索速度;

2、创建唯一性索引，保证数据库表中每一行数据的唯一性;

3、加速表和表之间的连接;

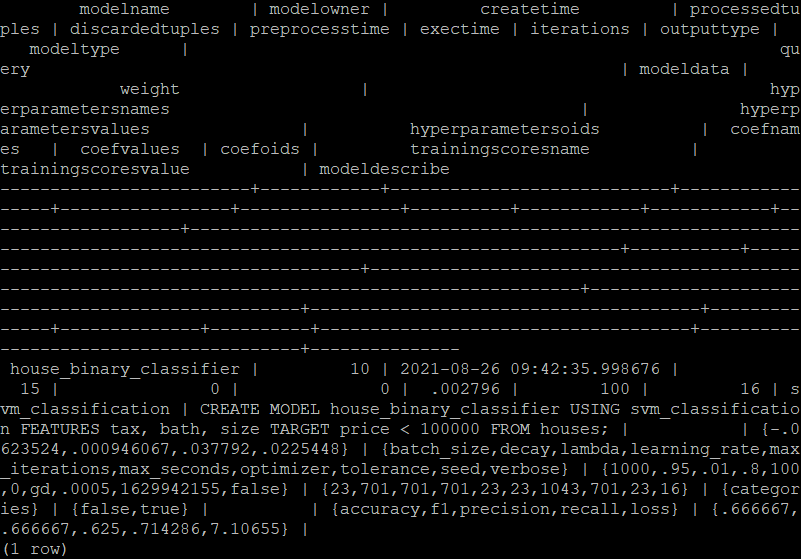
4、在使用分组和排序子句进行数据检索时，可以显著减少查询中分组和排序的时间。

选取最适用的字段属性；使用连接代替子查询；使用联合来代替手动创建的临时表；使用事务；使用外键；锁定表；优化查询语句。

# 关卡四、openGauss的DB4AI特性应用

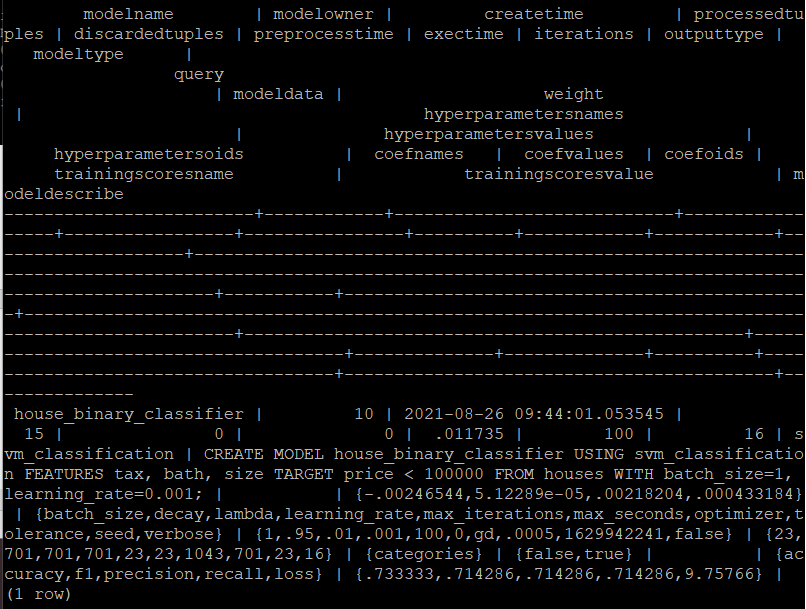
任务一：在gs\_model\_warehouse系统表中查看训练后的模型信息，将执行结果截图：

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



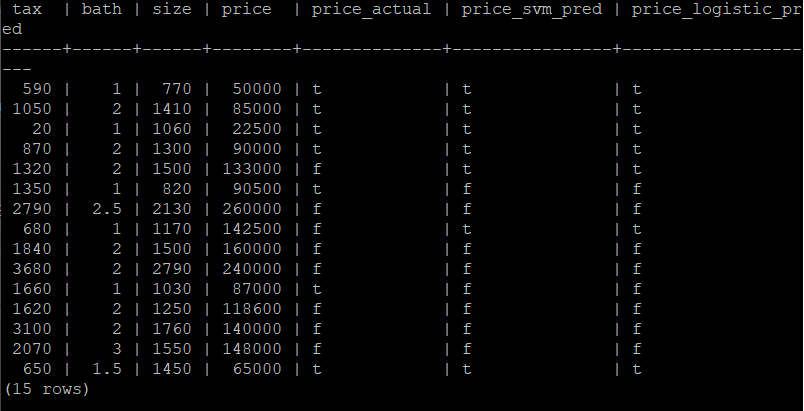
任务二：观察新模型的信息，将执行结果截图。

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？

分类和回归的区别在于输出变量的类型。定量输出称为回归，或者说是连续变量预测；

定性输出称为分类，或者说是离散变量预测。

实践思考题2：什么是SVM算法？

SVM(Support Vector Machine)指的是支持向量机，是常见的一种判别方法。在机器学习领域，是一个有监督的学习模型，通常用来进行模式识别、分类以及回归分析。

它是针对线性可分情况进行分析，对于线性不可分的情况，通过使用非线性映射算法将低维输入空间线性不可分的样本转化为高维特征空间使其线性可分，从而使得高维特征空间采用线性算法对样本的非线性特征进行线性分析成为可能。

它基于结构风险最小化理论之上在特征空间中构建最优超平面，使得学习器得到全局最优化，并且在整个样本空间的期望以某个概率满足一定上界。

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、错误率与精度

错误率及精度是分类任务中最常用的性能指标，适合于二分类及多分类任务，其定义也比较简单，错误率E指错误分类的样本数占总样本数的比例，精度则为1-E。

2、查准率、查全率

查准率P：在所有被标为“正例”的样本中，真正为“正例”的样本比例

https://private.codecogs.com/gif.latex?P%3D%5Cfrac%7BTP%7D%7BTP&plus;FP%7D

查全率R：在所有真正为“正例”的样本，被标记为“正例”的样本比例

https://private.codecogs.com/gif.latex?R%3D%5Cfrac%7BTP%7D%7BTP&plus;NP%7D

在实际应用中一般使用度量指标F1,它是基于查准率P与查全率R计算出来的，其定义如下：

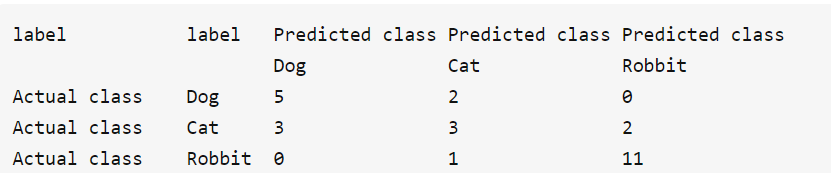
https://private.codecogs.com/gif.latex?F1%3D%5Cfrac%7B2%5Ctimes%20P%5Ctimes%20R%7D%7BP&plus;R%7D%3D%5Cfrac%7B2%5Ctimes%20TP%7D%7BN&plus;TP-TN%7D N为样例总数。

3、AUC

AUC是另一种评价二分类算法的指标，被定义为 ROC 曲线下的面积。受试者工作特征曲线 （receiver operating characteristic curve，简称ROC曲线）。ROC曲线的坐标，纵坐标为真正例率（True Positive Rate,TPR），横坐标为假正例率（False Positive Rate,FPR）。根据模型在ROC坐标上的位置，可以判断模型的性能。

4、混淆矩阵

对于多分类的相关问题，常用的评价标准是混淆矩阵。如果分成n个类别的话，混淆矩阵的形式是一个n\*n的一个矩阵。

例：

上图中每一列代表预测的类别，每一行代表真实的类别的归属，第二行cat的总数量为8，表示类cat总共有8个样本，其中3个分类正确，3个被错误的分到了dog，2个被错误的分到了robbit。

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？

回归的有：RMSE(平方根误差)、MAE（平均绝对误差）、MSE(平均平方误差)、Coefficient of determination (决定系数)。