openGauss 数据库维护管理实验

姓名： 周钰宸 学号： 2111408

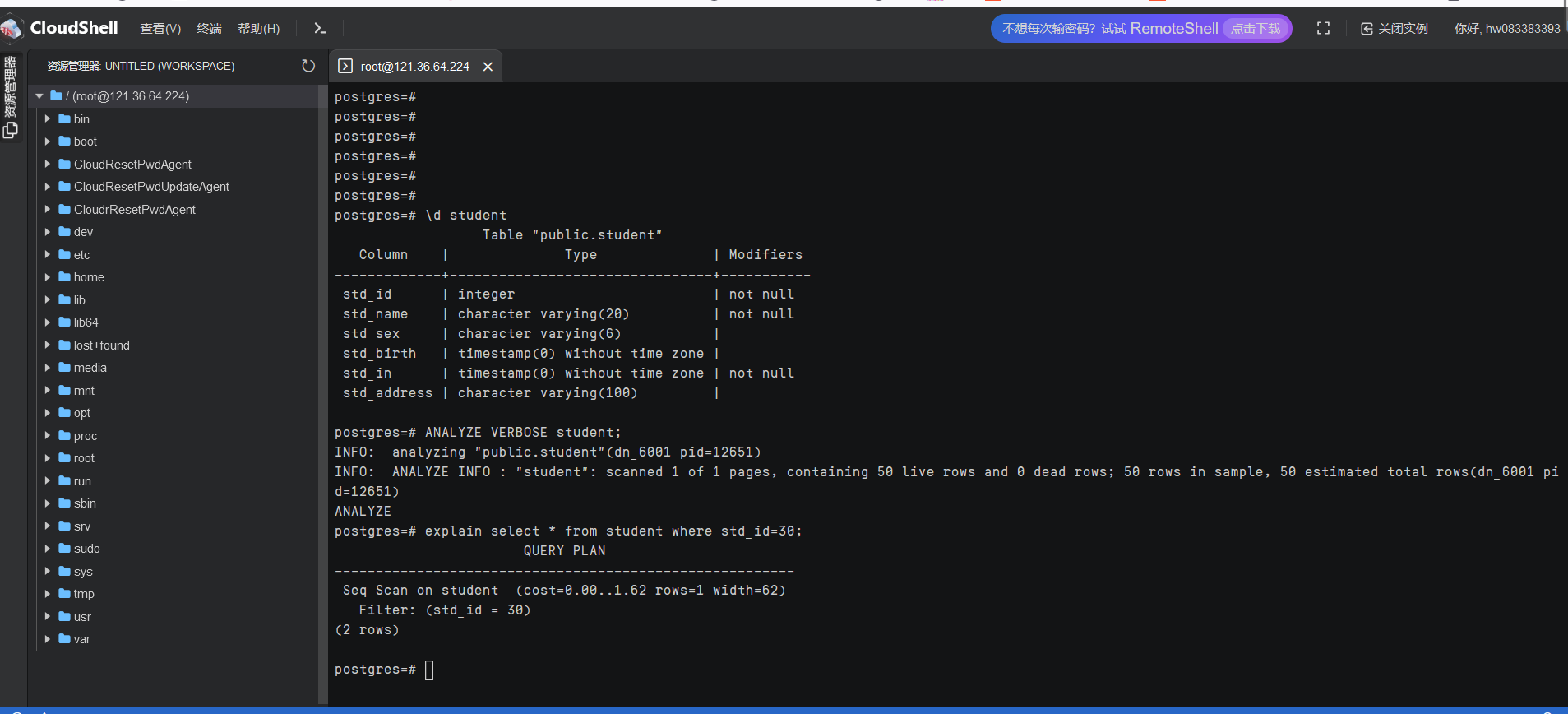
实验步骤：

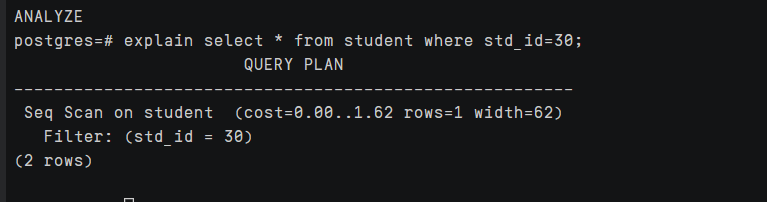
* openGauss数据库安装
* 数据库性能检查实验
* 最大连接数设置实验
* 例行表、索引维护实验

实验报告

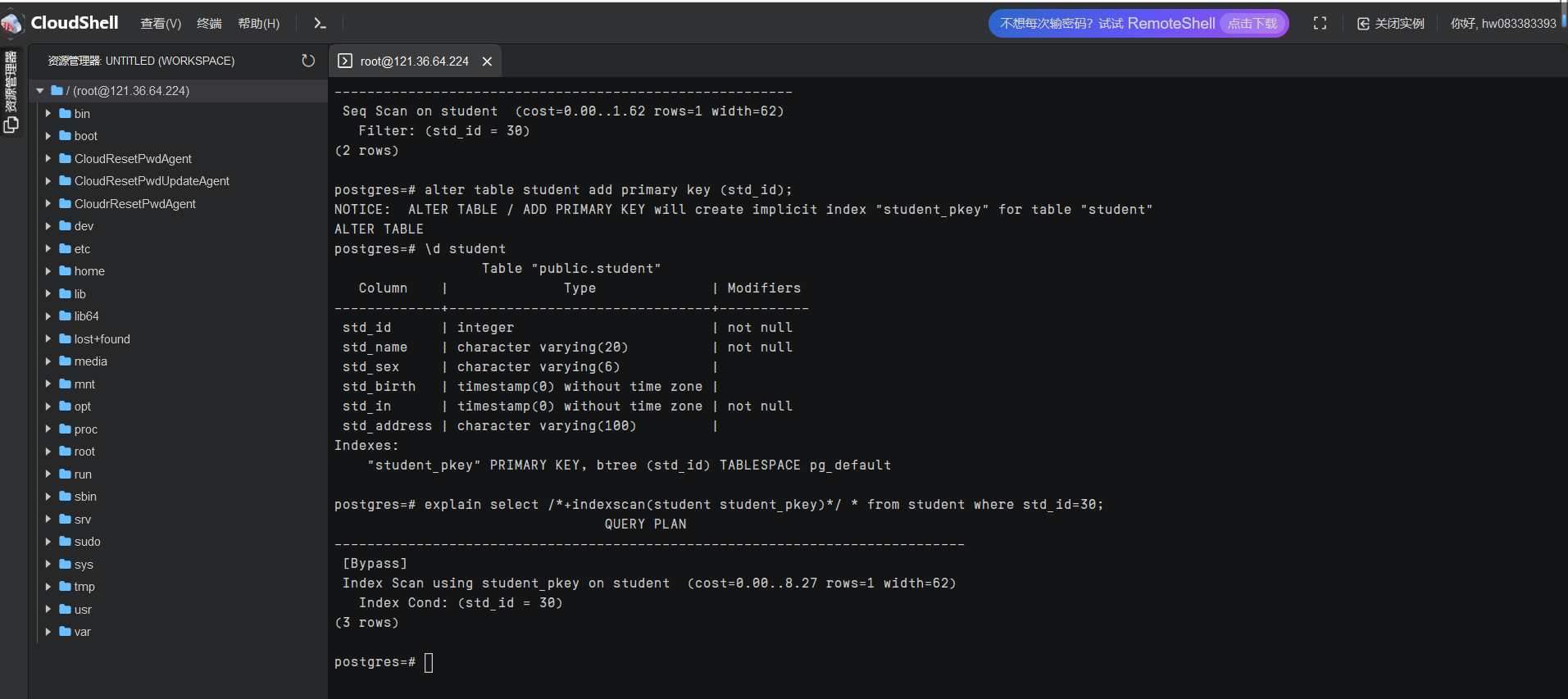
实验步骤截图：

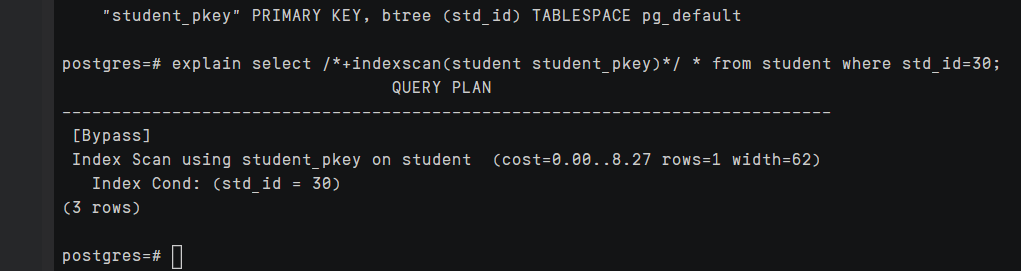
截图1：指导手册25页顺序扫描执行计划截图



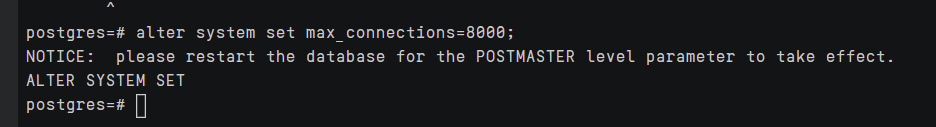


截图2：指导手册26页索引扫描执行计划截图

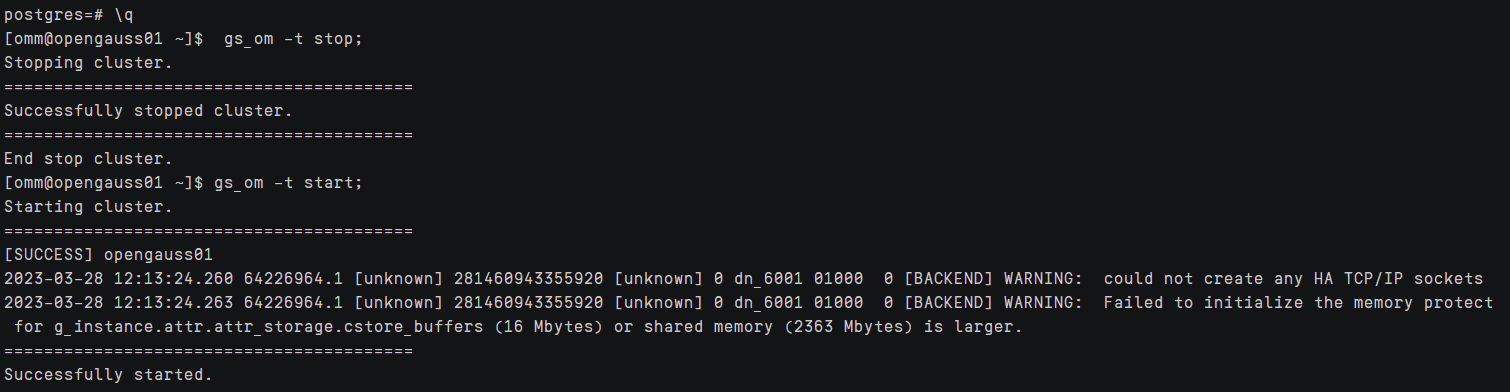




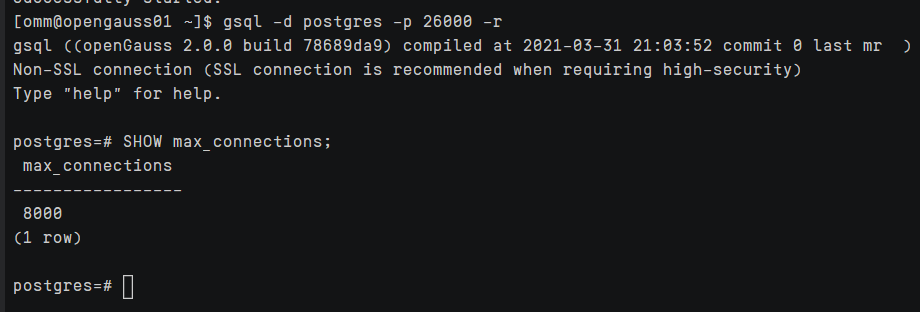
截图3：将最大连接数设置为8000并验证设置是否成功截图（注意，指导手册中将最大连接数设置为6000，怎样重新设置为8000呢？）



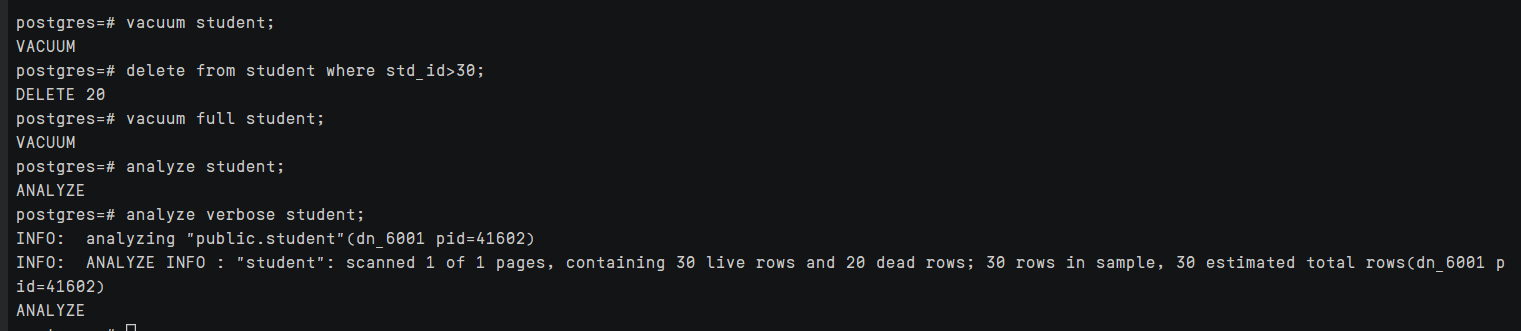
通过将**alter system set max\_connections=6000;**中的”6000”改为”8000”即可，如上图所示。接下来验证设置的截图如下：

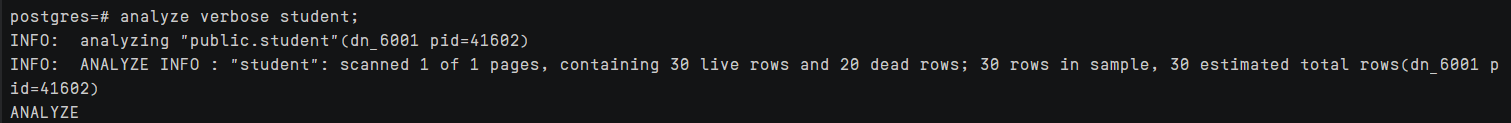


(然后先退出，并重启数据库，如上图所示）



**（使用gsql客户端以管理员用户身份连接postgres数据库，然后查看参数值。可以看到最大连接数为8000，故验证成功）**

截图4：使用ANALYZE VERBOSE语句更新统计信息，并输出表的相关信息。



(由于上图较小，所以这里把这句所生成的代码信息显示在下面——**也可以放大查看上图结果**）

**postgres=# analyze verbose student;**

**INFO: analyzing "public.student"(dn\_6001 pid=41602)**

**INFO: ANALYZE INFO : "student": scanned 1 of 1 pages, containing 30 live rows and 20 dead rows; 30 rows in sample, 30 estimated total rows(dn\_6001 pid=41602)**

**ANALYZE**

实验思考题：

1. 全表扫描和索引扫描的区别是什么？具体是如何实现的？比较两种扫描方式的cost（提供查询结果截图）,为什么全表扫描比索引扫描cost更小？在什么情况下通过主键进行查找会比全表扫描更节省时间？

**（1）区别：全表扫描和索引扫描都是数据库中常用的查询方式——**

①全表扫描：指在不使用索引的情况下，对一张表中的每一行都依次进行遍历，逐行地检查是否符合查询的条件。这方方式一般情况下需要耗费大量的时间和资源，因此只适用于查询条件无法利用索引或小数据集的情况。

②索引扫描：指在使用索引的情况下，查找表中哪个行符合条件，而不是直接i去遍历整个表。索引扫描只需要检索部分数据，与全表扫描相比，效率要高出很多。但是如果非索引列出现在了查询条件中，那么仍需要采用全表扫描。

总之，**索引扫通过使用索引，来加快查询速度，但是需要针对具体查询条件进行索引优化；而全表扫描则适用于较小的数据集，或者查询条件无法使用索引的情况。**

1. **具体如何实现——**

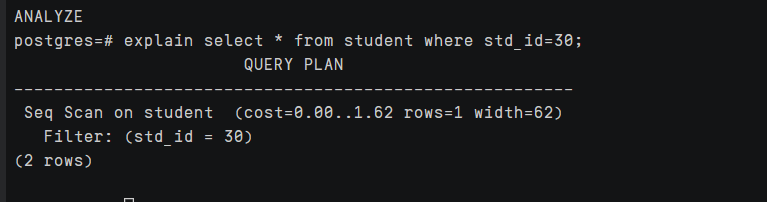
①全表扫描：通过逐一读取磁盘上存储的数据页，将其中的每一条记录都读入内存，再进行条件判断筛选、排序等操作，需要耗费大量的磁盘I/O操作。

②索引扫描：根据已有的查询条件，到索引的B-tree结构进行快速查找，寻找符合条件的的记录，然后通过索引指针读取相应的数据行。

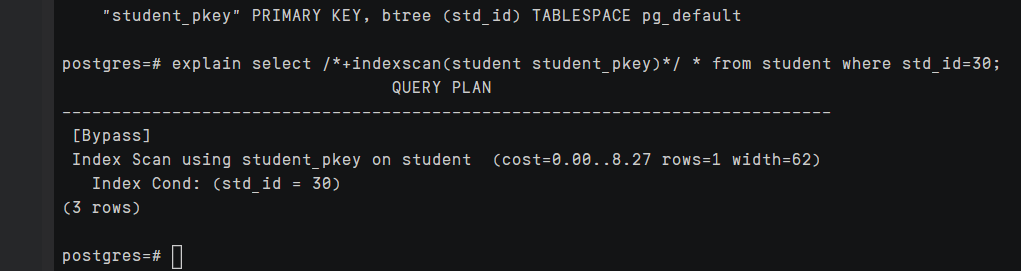
1. **比较两种扫描方式的cost——**

**①查询结果在上图实验步骤部分的截图1和截图2。**

**全表扫描：**



**索引扫描：**



**可以通过比较发现，全表扫描的cost=0.00..1.62＜索引扫描的 cost=0.00..8.27。**

**②为什么全表扫描与索引扫描cost更小：**

这是因为在某些情况下，全表扫描比索引扫描更快，并且在某些情况下，即使使用索引扫描也需要访问大部分表数据，所以代价较高。

首先，若所要查询的表非常小，即整个表的大小可以被内存所容纳时，全表扫描通常比索引扫描更快。这是由于若查询条件不够严格，**索引扫描需要去访问索引的B-tree，花费额外的时间来查找并读取每个符合记录的位置，而对于较小的表，这些额外的开销可能会成为瓶颈。而全表扫描没有额外的用来访问索引B-tree的开销和来自内存缓存的开销。**

其次，在某些情况下，**查询涉及到了表的大部分数据，甚至超过了索引扫描的覆盖范围。**

最后，**索引扫描可能会产生很多随机的I/O操作，而全表扫描则更倾向于顺序的I/O操作**。在某些情况下，顺序的I/O的代价可能比随机的I/O更小，因为它可以更好地利用磁盘的预读取能力。

1. **在什么情况下通过主键进行查找会比全表扫描更节省时间——**

当查询条件中包含了表的主键或唯一索引时。这是因为主键或唯一索引可以通过索引扫描直接定位到需要查询的数据行，而不需要扫描整个表，此时索引扫描的cost会比全表扫描低，即更节省时间。同时，使用主键或唯一索引进行查找还可以避免重复记录的情况。

1. 请列举一种需要重建索引的情况和原因，并说明openGauss中重建索引的方式有哪些。
2. **一种需要重建索引的情况和原因——索引过度碎片化**

当索引被频繁更新或删除时，可能会导致索引碎片化。这会使查询变慢，因为数据库需要花费更多的时间来查找数据。在这种情况下，重建索引可以解决碎片化问题，提高查询性能。

1. **openGauss中重建索引的方式——**

**①使用CREATE INDEX命令重新创建索引**：可以使用CREATE INDEX命令来重新创建一个已经存在的索引。如果该索引已经存在，则会先删除旧索引再重新创建新索引。这个过程可以在不影响表的读写操作的情况下进行。

**②使用REINDEX命令重建索引：**REINDEX命令会重新生成指定索引的数据文件，并更新系统表中该索引的元数据信息。这个过程需要对表进行独占锁定，因此可能会对表的读写操作造成一定的影响。