openGauss 数据库维护管理实验

姓名： 冯思程 学号： 2112213

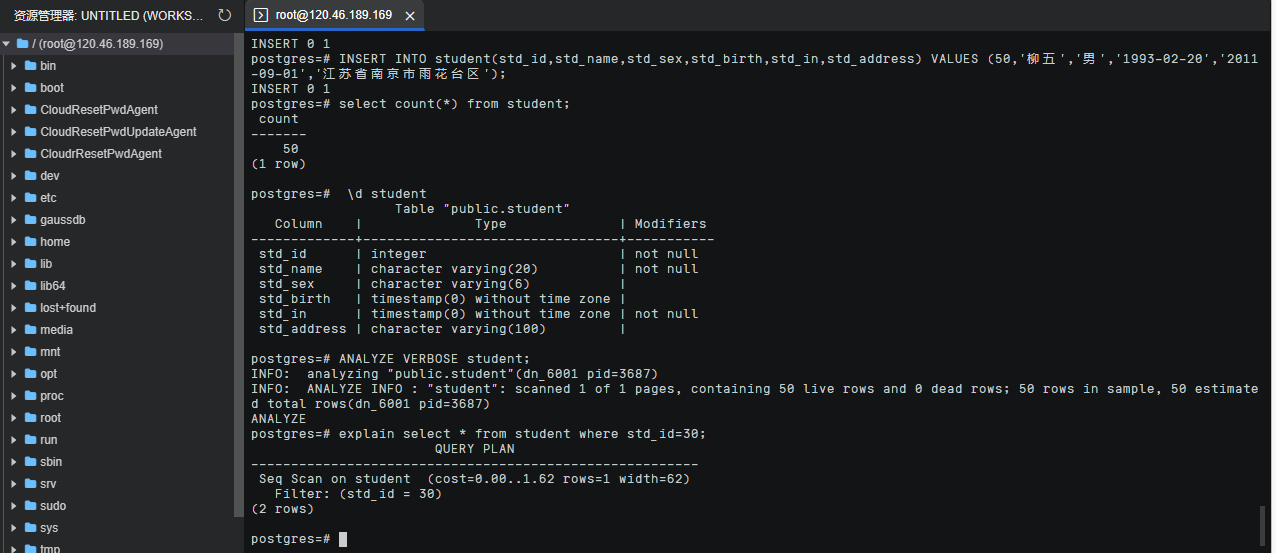
实验步骤：

* 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

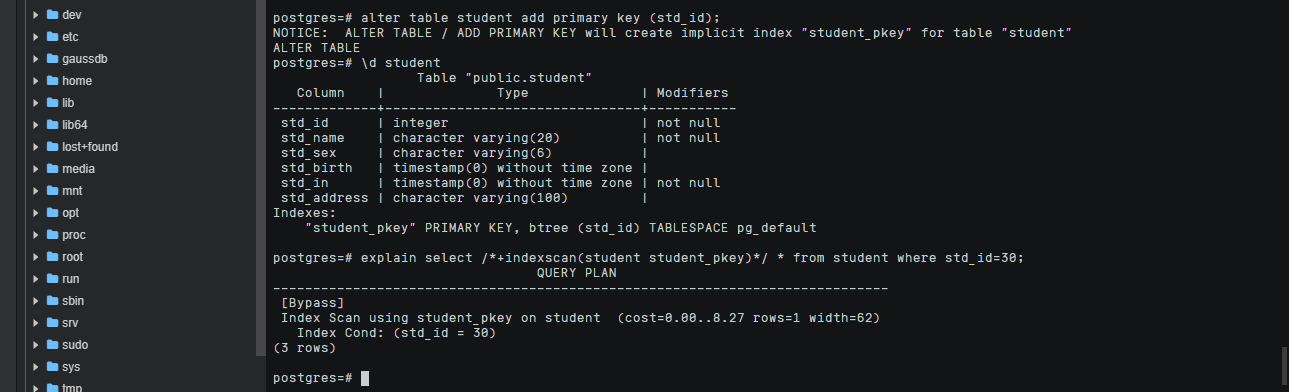
实验报告

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？）

设置前最大连接数截图：



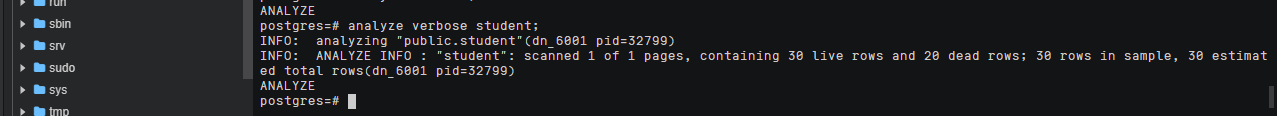
重新设置最大连接数后更新数据库再次观测最大连接数截图：



对于问题的回答：

在用alter system set语句来调整最大连接参数时候，将6000改成8000，然后执行指令。

截图4：在进行例行表、索引的维护实验中，使用ANALYZE VERBOSE语句更新统计信息，并输出表的相关信息截图。



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？

答：全表扫描和索引扫描的区别在于多块读取和单独读取的方式。

1、全表扫描：为实现全表扫描，读取表中所有的行，并检查每一行是否满足语句的限制条件。利用多块读的方法可以十分高效地实现全表扫描，而且只有在全表扫描的情况下才能使用多块读操作。

2、索引扫描：我们先通过查找到数据对应的值，然后直接从表中得到具体的数据，这种查找方式称为索引扫描。

根据截图可以知道，全表扫描的cost是0.00..1.62，索引扫描的cost是0.00..8.27，全表的cost更小。原因：首先我们知道较大量的数据存储会存储在很多个数据块中，当我们进行全表扫描的时候，我们是多个块一起读取的，不需要重复访问块，而当我们建立索引进行索引扫描的时候，读取是分块读取的，每一次读取都要去饭访问一次目标数据块。很明显，在进行大量数据的读取时候，重复访问数据块会大程度拖慢读取效率，索引在本次实验中，全表读取的cost小于索引读取的cost。

在下面情况下使用主键查找会比全表扫描更快：

1. 只使用主键索引就能够返回查询结果的查询，比如聚合查询中的函数count，max，min。无需去挨个访问数据块来读取。
2. 查询结果的记录数量小于表中记录一定比例的时候。根据上面提到全表扫描和索引扫描的读取方式区别，可以看出当我们通过索引需要访问的记录足够少的时候，扫描效率就会高于全表扫描了。
3. 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。

答：

1. ：

原因一：如果数据发生大量删除后，索引页面上的索引键将被删除，导致索引页面数量的减少，造成索引膨胀。重建索引可回收浪费的空间。

原因二：新建的索引中逻辑结构相邻的页面，通常在物理结构中也是相邻的，所以一个新建的索引比更新了多次的索引访问速度要快。

1. ：重建索引有以下两种方式：
2. 使用REINDEX语句重建索引，示例如下：

postgres=# **reindex table student;**

REINDEX

postgres=#

1. 先删除索引（DROP INDEX），再创建索引（CREATE INDEX）。示例如下：

postgres=# **drop index inx\_stu01;**

DROP INDEX

postgres=# **create index inx\_stu01 on student(std\_name);**

CREATE INDEX

postgres=#