openGauss 数据库开发查询实验

姓名： 孙蕗 学号： 2112060

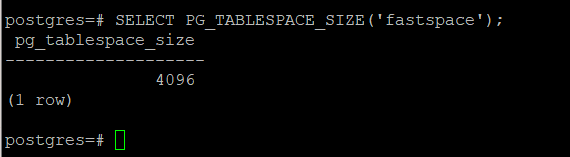
实验步骤：

* 创建和管理用户、表空间和数据库
* 创建和管理表
* 创建和管理其他数据库对象
* 学校数据模型创建及表操作

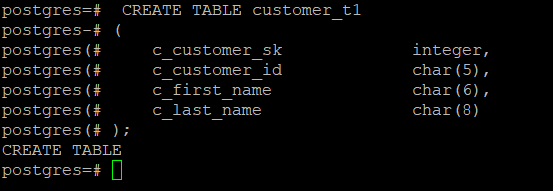
实验报告

实验步骤截图：

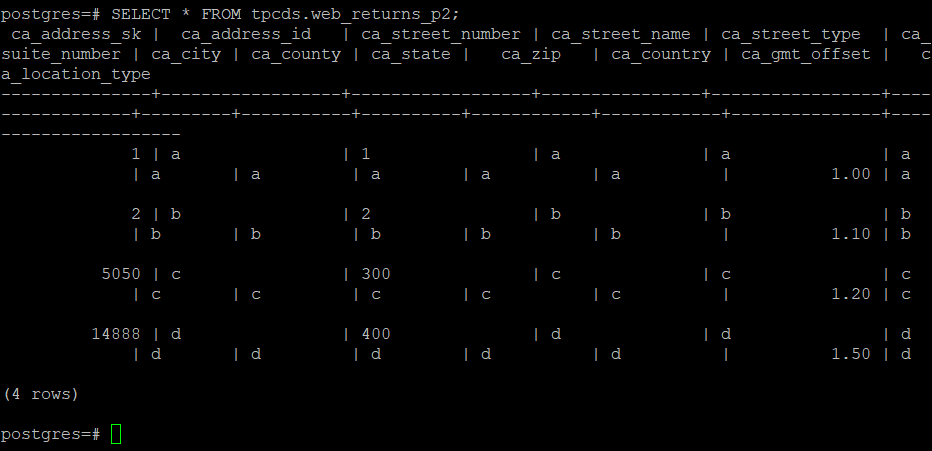
截图1：指导手册第8页，查询表空间当前使用情况截图



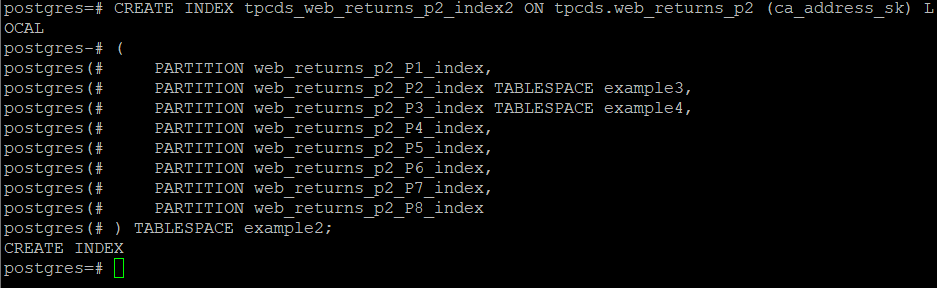
截图2：指导手册第10页，创建表截图



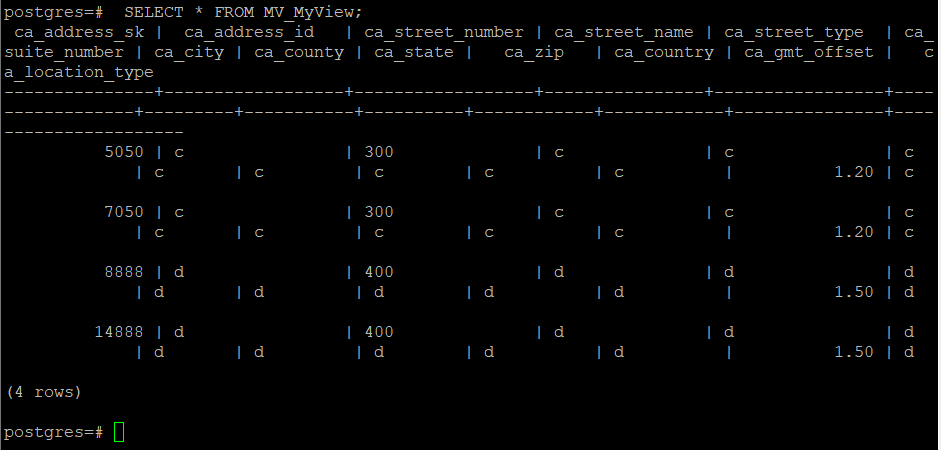
截图3：指导手册第16页，向分区表中插入数据后查看分区表中所有数据并截图（该命令需自行撰写）



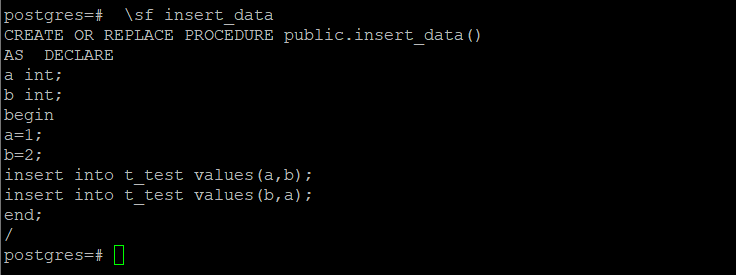
截图4：指导手册第19页，创建分区索引截图。



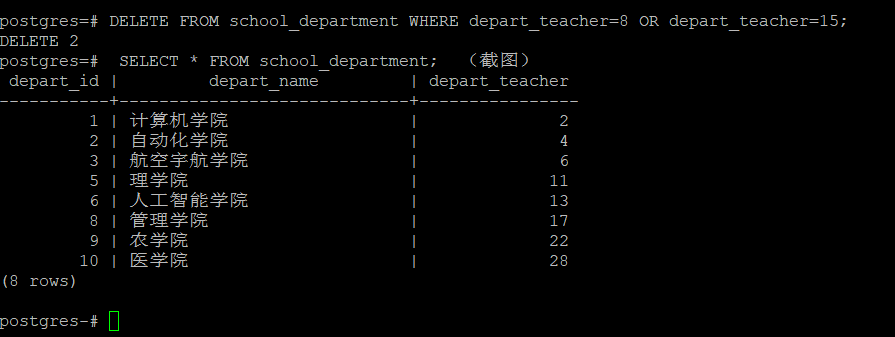
截图5：指导手册第23页，更新物化视图。



截图6：指导手册第26页，管理存储过程



截图7：指导手册第39页，删除数据后表中内容截图



实验思考题：

1. 在openGauss中，创建具有“创建数据库”权限的用户Alice，并设置其初始密码为”openGauss@0331”,应使用的语句是：

答： CREATE USER Alice CREATEDB PASSWORD 'openGauss@0331';

1. 命令 “DROP USER kim CASCADE”的效果是？（可以预习参考第八周主讲课内容，权限和授权）

答：删除用户，将永久删除所有用户的数据和对象，并且不可恢复。

这个命令是用来从数据库中删除用户 "kim" 的，CASCADE选项表示在删除该用户的同时，也会删除该用户拥有的所有对象，例如表、视图、函数、存储过程等等。

如果用户 "kim" 没有创建任何对象，那么在删除该用户时，CASCADE选项的效果将是没有任何影响。

1. 向表中插入数据时，是否允许只对部分属性插入数值？在何种情况下允许，应如何书写语句？何种情况下不允许？

答：在关系型数据库中，向表中插入数据时允许只对部分属性插入数值，这被称为部分插入（Partial Insertion）。允许部分插入的情况主要包括以下两种：

1. 表中存在默认值：当某个属性在表定义时指定了默认值，如果不对该属性进行赋值，则该属性的默认值会被自动赋予。在这种情况下，可以只对需要赋值的属性进行赋值。

例如，假设有一个名为“user”的表，其中包含“id”、“name”、“age”和“gender”四个属性，其中“gender”属性指定了默认值为“unknown”。则以下语句是合法的：

INSERT INTO user(id, name, age) VALUES (1, 'Tom', 18);

1. 插入语句中使用了空值符号：在插入语句中，可以使用空值符号（NULL）表示某个属性的值为空。在这种情况下，可以只对需要赋值的属性进行赋值，未赋值的属性会被自动赋值为空值符号。

例如，假设有一个名为“order”的表，其中包含“id”、“user\_id”、“product\_id”和“quantity”四个属性，其中“user\_id”属性可以为空。则以下语句是合法的：

INSERT INTO order(id, product\_id, quantity) VALUES (1, 1001, 2);

在某些情况下不允许只对部分属性插入数值，例如：

1. 表中所有属性都没有默认值，也没有使用空值符号，此时必须对所有属性进行赋值。
2. 表中某些属性定义了非空约束（NOT NULL），此时必须对这些属性进行赋值，否则会导致插入失败。
3. 是否可以向表中一次性插入多条数据？何种插入效率较高？

答：可以

1. 使用多个INSERT INTO语句分别插入不同的数据，每个INSERT INTO语句只插入一条数据。这种方式的优点是灵活性较高，可以根据需要插入不同的数据，缺点是插入效率较低，因为每个插入语句都需要单独执行一次数据库操作。

例如，以下语句可以插入三条数据：

INSERT INTO user(id, name, age) VALUES (1, 'Tom', 18);

INSERT INTO user(id, name, age) VALUES (2, 'Jerry', 20);

INSERT INTO user(id, name, age) VALUES (3, 'Maggie', 25);

1. 使用单个批量插入语句：可以使用单个INSERT INTO语句一次性插入多条数据，这种方式的优点是插入效率较高，因为可以将多个数据一次性提交到数据库进行处理。缺点是语句的书写比较复杂，需要使用特殊的语法来指定要插入的数据。

例如，以下语句可以使用VALUES子句一次性插入三条数据：

INSERT INTO user(id, name, age)

VALUES (1, 'Tom', 18), (2, 'Jerry', 20), (3, 'Maggie', 25);

1. openGauss中将表中所有元组删除的两种命令是？

答：

1. TRUNCATE命令：该命令可以快速删除表中所有数据，但不会删除表的结构。TRUNCATE命令不需要WHERE子句，它会直接删除整个表中的所有数据，并且不会触发触发器。

TRUNCATE TABLE table\_name;

table\_name为要删除数据的表名。

删除名为mytable的表中的所有数据：TRUNCATE TABLE mytable;

1. DELETE命令：该命令可以通过WHERE子句指定要删除的数据，也可以不使用WHERE子句删除所有数据。DELETE命令可以触发触发器，在删除数据时可以执行其他的操作，例如删除相关的子表数据等。

DELETE FROM table\_name [WHERE condition];

table\_name为要删除数据的表名，condition为WHERE子句中的删除条件。如果不使用WHERE子句，则删除表中所有数据。

删除名为mytable的表中的所有数据：DELETE FROM mytable;

TRUNCATE和DELETE命令都可以删除表中的所有数据，但它们的实现方式不同，TRUNCATE命令会比DELETE命令快一些，但DELETE命令更加灵活，可以根据需要指定删除条件。

1. 如果经常需要查询某字段值小于某一指定值的信息，可以如何操作？（提示，从索引角度思考）

答： 如果经常需要查询某字段值小于某一指定值的信息，可以通过创建索引来提高查询效率。索引是一种特殊的数据结构，可以按照指定的字段值对表中的数据进行排序和组织，从而提高查询效率。

可以使用B-tree索引来加速对某字段值小于指定值的查询。B-tree索引是一种广泛使用的索引类型，它可以对索引字段进行排序，并支持范围查询和快速定位。

在创建B-tree索引时，可以指定要索引的字段名和排序方式。如果需要对某个字段进行小于查询，可以使用升序排序方式来创建索引。这样，在查询时，数据库系统就可以利用索引进行范围查询，从而快速定位满足条件的数据。

1. 在什么场景下可以使用物化视图？物化视图和普通视图的区别是？

答：物化视图是一种数据库对象，它可以存储一个查询结果的快照，并且可以在查询时直接使用该快照，而不需要每次都重新计算。物化视图可以极大地提高查询性能，特别是对于复杂查询或聚合查询等高开销的操作。

物化视图和普通视图的区别如下：

存储方式不同：普通视图不存储任何数据，它只是一个查询的逻辑表示。而物化视图实际上存储了一个查询结果的快照，可以直接使用该快照进行查询，而不需要每次都重新计算。

更新方式不同：普通视图无法更新，它只能用于查询。而物化视图可以在需要时更新快照，以保持其与源表的同步。

查询效率不同：普通视图每次查询都需要重新计算，因此在查询大量数据或者需要进行复杂计算的情况下，查询性能可能会较低。而物化视图可以直接使用快照进行查询，因此查询性能更高，尤其是对于复杂查询或聚合查询等高开销的操作。

在实际应用中，可以在以下场景下使用物化视图：

对于复杂查询或聚合查询等高开销的操作，可以使用物化视图来提高查询性能。

对于需要实时查询的业务场景，可以使用物化视图来缓存查询结果，以提高查询性能。

对于数据仓库或BI系统等需要进行复杂计算的场景，可以使用物化视图来加速查询。

物化视图虽然可以提高查询性能，但同时也会增加存储开销和维护成本。因此，在使用物化视图时需要权衡存储和性能的关系，并根据具体业务场景进行优化。

1. 学校模型ER图绘制

