openGauss 数据库开发查询实验

姓名： 周钰宸 学号： 2111408

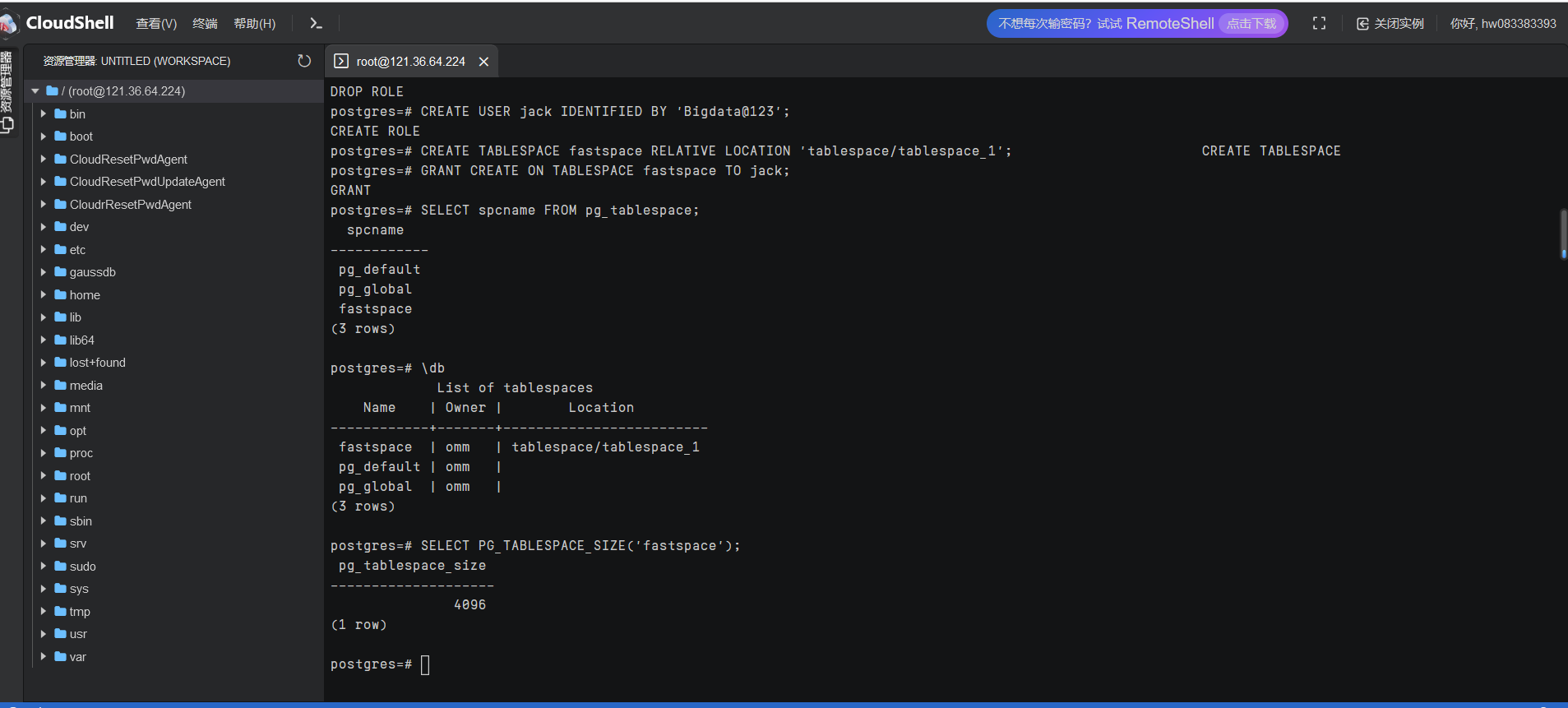
实验步骤：

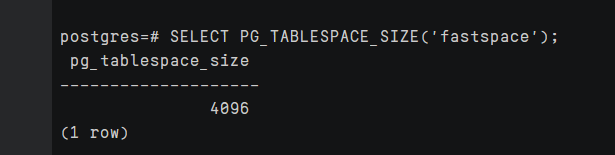
* 创建和管理用户、表空间和数据库
* 创建和管理表
* 创建和管理其他数据库对象
* 学校数据模型创建及表操作

实验报告

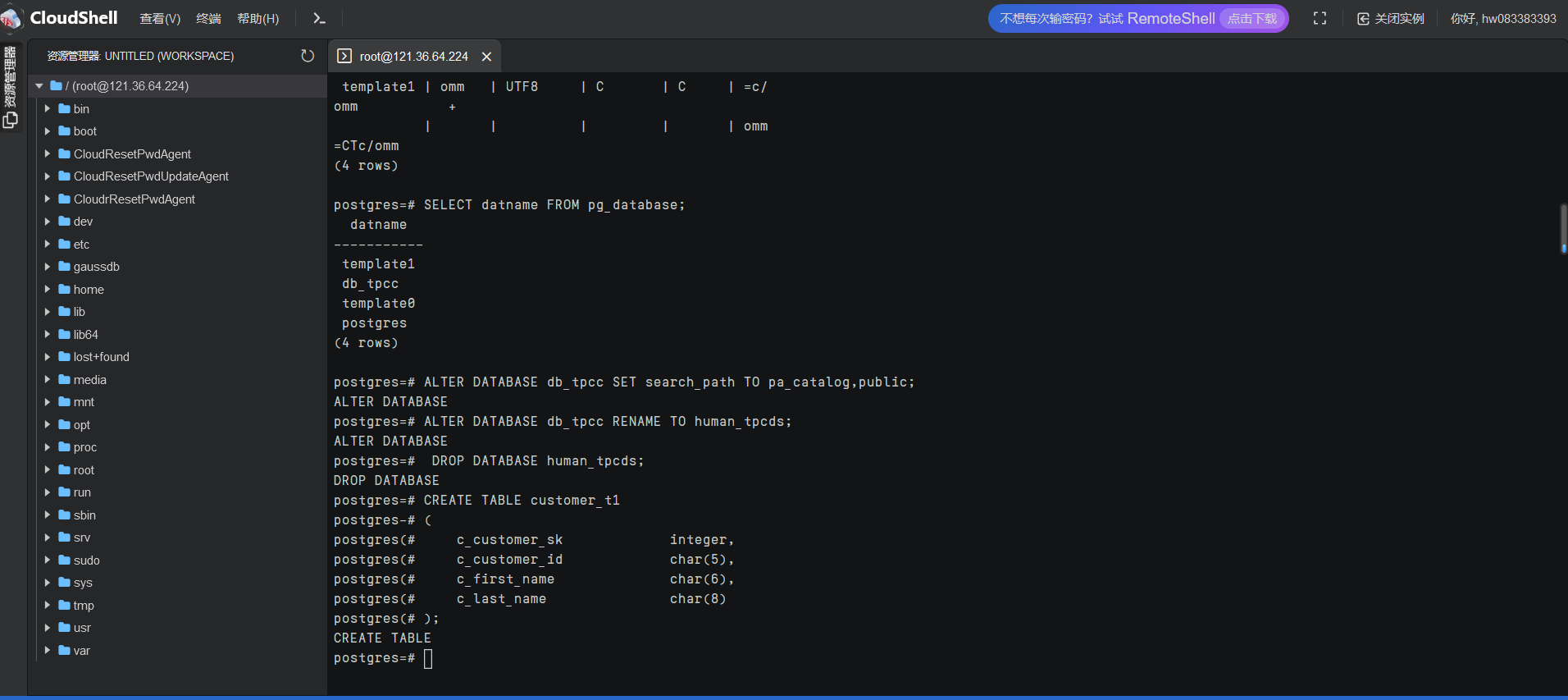
实验步骤截图：

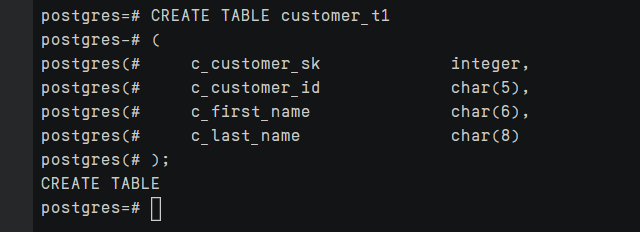
截图1：指导手册第8页，查询表空间当前使用情况截图



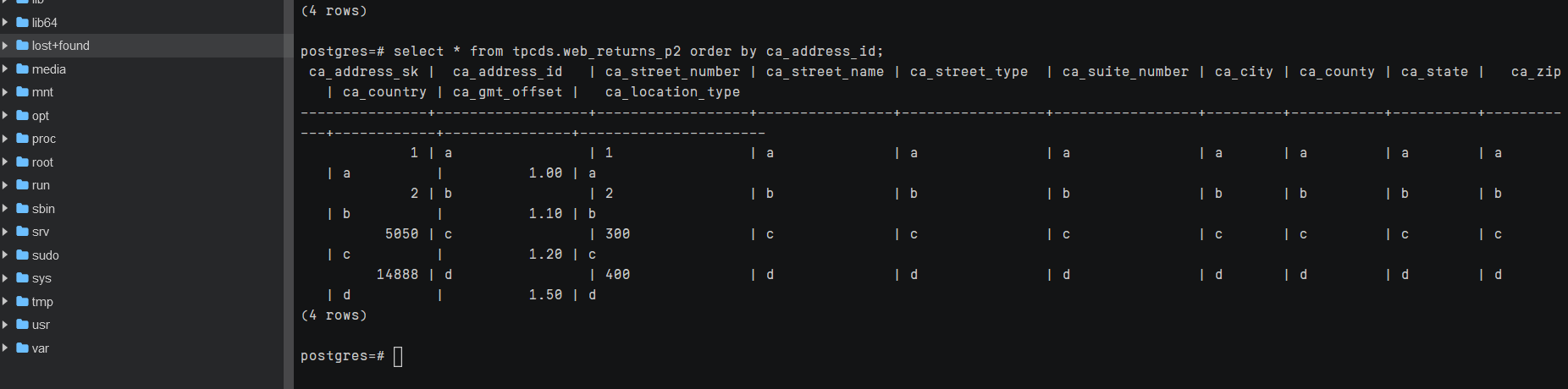


截图2：指导手册第10页，创建表截图





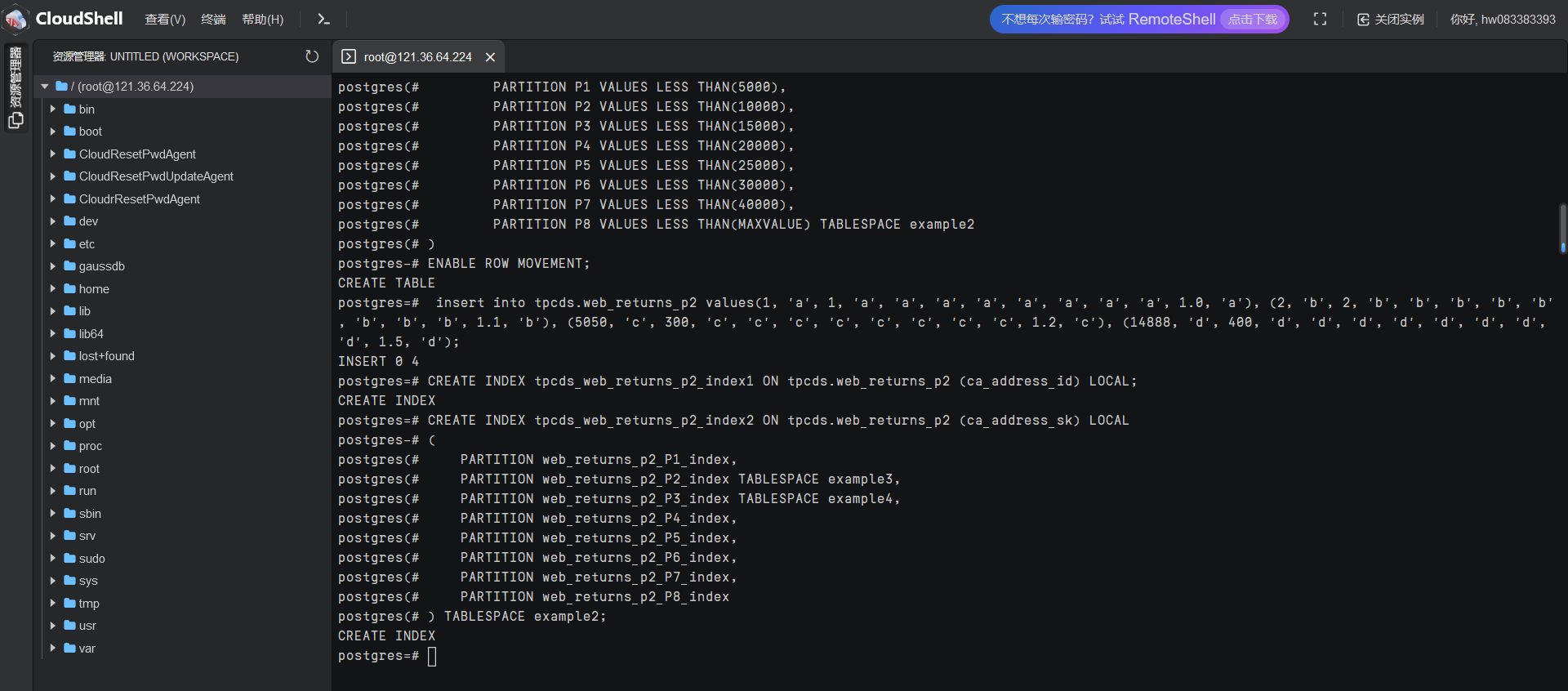
截图3：指导手册第16页，向分区表中插入数据后查看分区表中所有数据并截图（该命令需自行撰写）

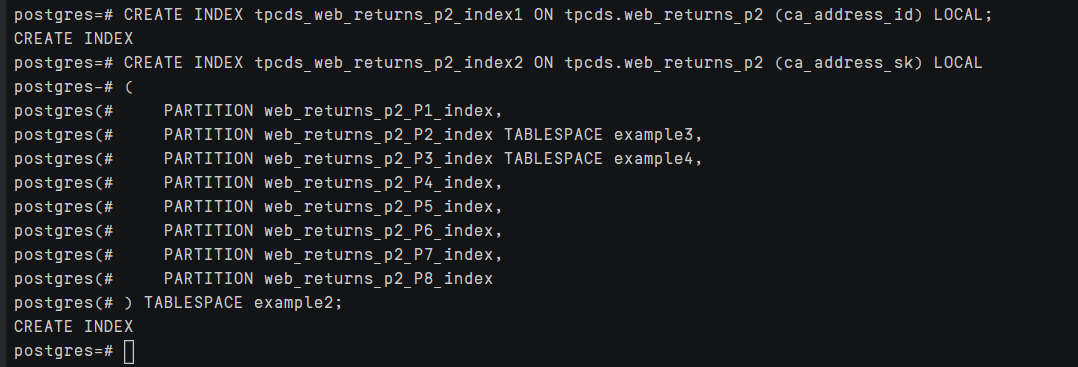


**对应命令为select \* from tpcds.web\_returns\_p2 order by ca\_address\_id;**

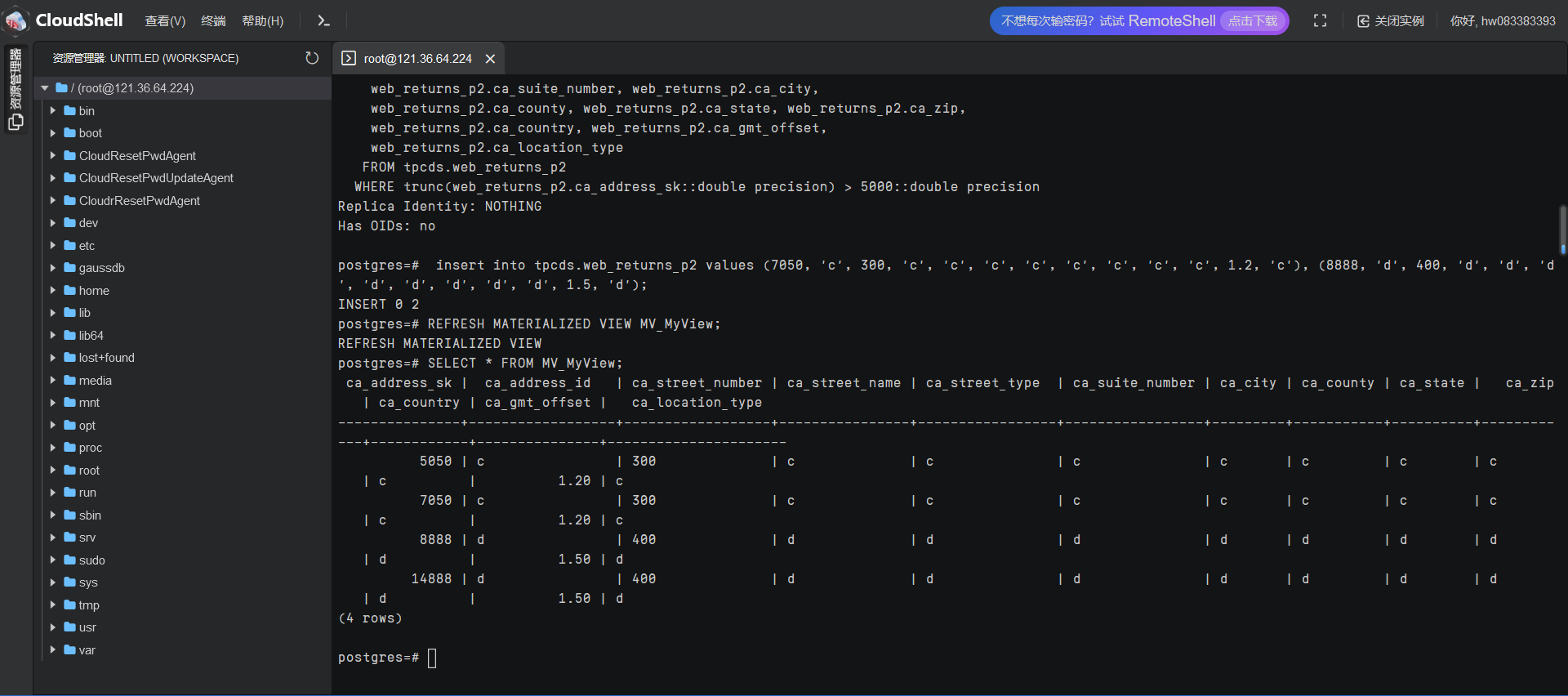
（即从tpcds.web\_returns\_p2表中查询所有数据，并且将数据按照ca\_address\_id排序的方式展示。）

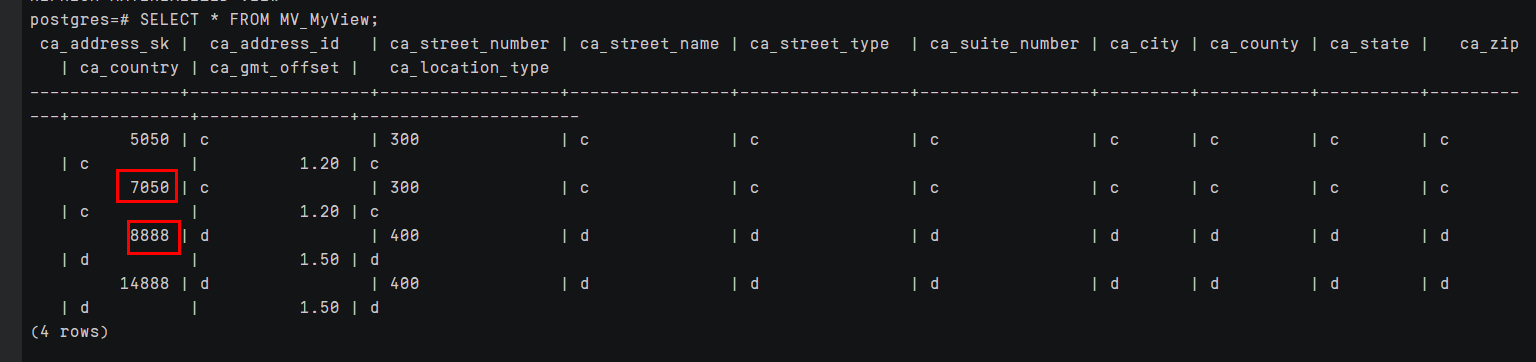
截图4：指导手册第19页，创建分区索引截图。



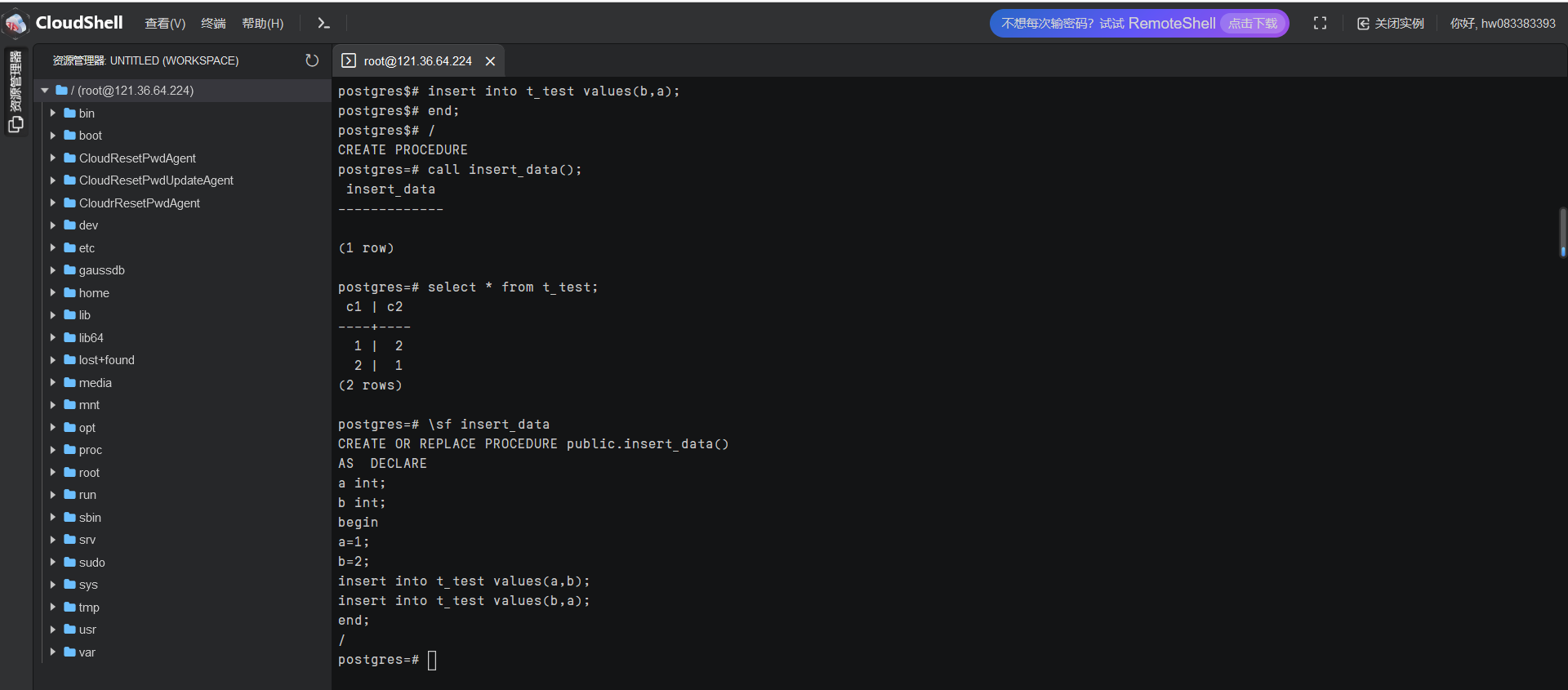


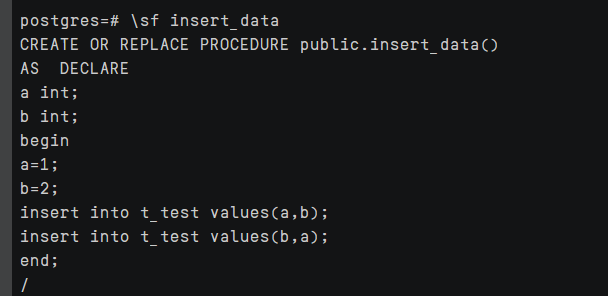
截图5：指导手册第23页，更新物化视图。



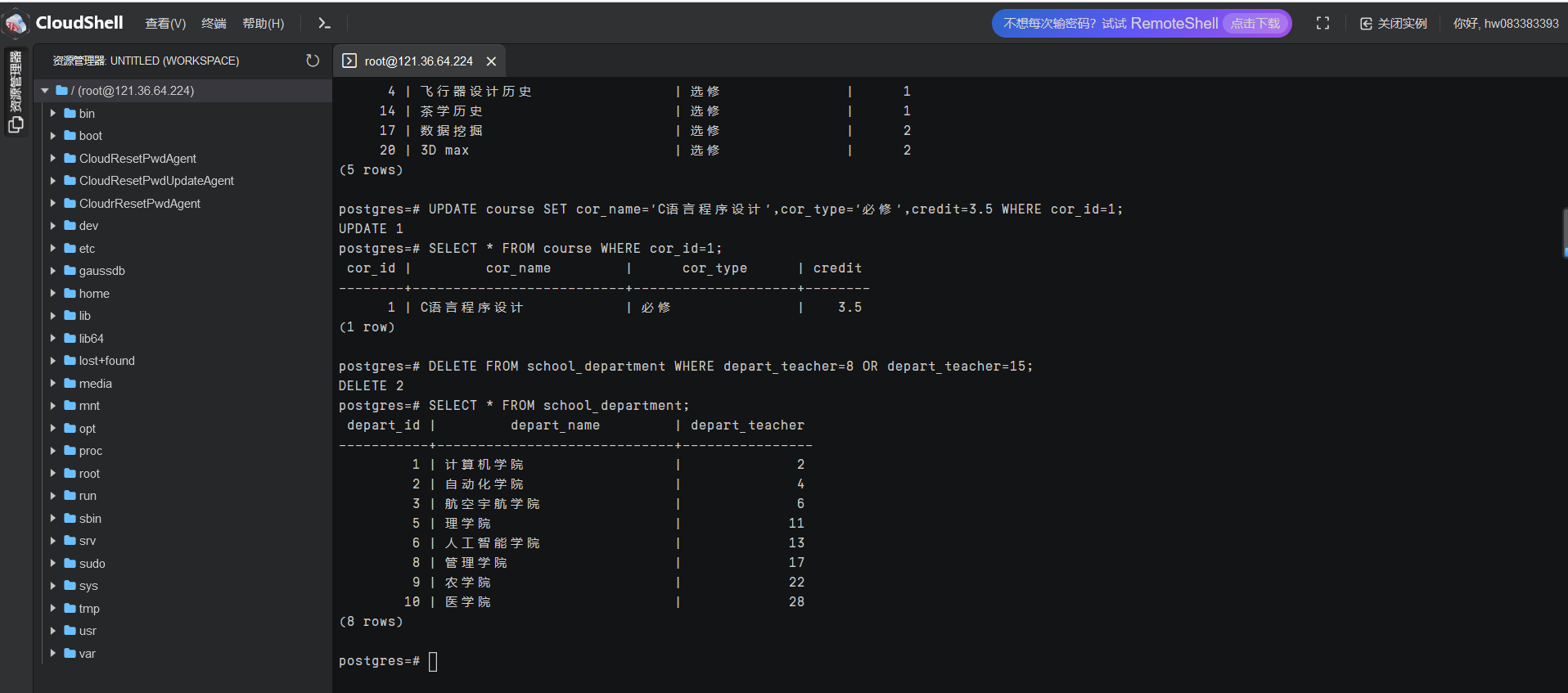


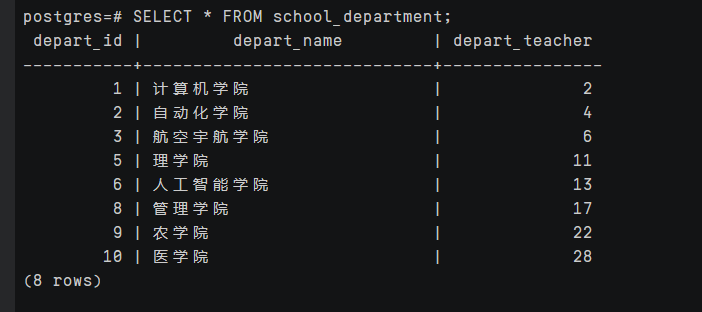
截图6：指导手册第26页，管理存储过程





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：**

**Create USER Alice 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：DROP USER kim CASCADE命令用于删除数据库中的一个用户kim以及与其相关联的所有对象。当使用CASCADE参数时，该命令将级联删除用户与该用户相关的所有数据库对象，包括该用户创建的表、视图和其他数据库对象。如果没有指定CASCADE参数，则删除操作将失败，因为该用户拥有其他对象。

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：在某些情况下，可以只对部分属性插入数值。例如，当一个表有很多列，但是你只想为其中的一些列插入数据时，可以只给这些列指定数值，而不考虑其他列。此外，在某些情况下，一些列可能已经配置为自动为其生成默认值，因此可以省略对这些列的数值输入。

要向表中仅插入某些列的值，可以使用 INSERT INTO 语句的语法来指定要插入的列名和对应的值。例如，如果我们只想插入 column1 和 column2 的值，可以使用以下语句：

INSERT INTO table\_name (column1, column2) VALUES (value1, value2);

在此示例中，table\_name 是要插入数据的表名，而 column1 和 column2 是要插入值的列名。value1 和 value2 是要插入的到 column1 和 column2 中的具体值。如果我们想同时插入三个列的值，可以使用以下语句：

INSERT INTO table\_name (column1, column2, column3) VALUES (value1, value2, value3);

在此示例中，我们使用了与前面相同的 INSERT INTO 语句的语法，只不过这里我们指定了表 table\_name 中所有三个列的值。无论哪种情况，我们都必须确保列名和值的数量、类型和顺序一致，以避免出现语法错误或数据不一致的问题。另外，如果表已经定义了默认值，则只需要插入需要的列即可，剩余列默认使用默认值。

1. **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：可以向表中一次性插入多条数据。这种操作通常使用 INSERT INTO 语句来完成。

在 SQL 中，可以使用以下语法向表中一次性插入多条数据：

INSERT INTO table\_name (column1, column2, ...) VALUES

(value1\_1, value2\_1, ...),

(value1\_2, value2\_2, ...),

...

(value1\_n, value2\_n, ...);

在这里，table\_name 是要插入数据的表名，column1、 column2 等是要插入数据的列名，而 value1\_1、 value2\_1 等则是对应的第一行要插入的数值，value1\_2、 value2\_2 等则是对应的第二行要插入的数值，以此类推。可以在括号内添加更多的值来一次性插入多条数据。

在进行大量数据插入时，批量插入比逐行插入效率更高。因为在批量插入时，数据库只需执行一次插入操作，而不是执行多个插入操作。因此，如果需要插入大量数据，则建议使用批量插入的方式。同时，在进行批量插入时，可以通过减少索引、关闭约束、预分配日志空间等技术手段来进一步提高插入效率。

1. **openGauss中将表中所有元组删除的两种命令是？**

在 openGauss 中，可以使用以下两种命令将表中所有元组删除：

（1）TRUNCATE TABLE：该命令会快速地删除表中的所有数据，但是不会产生回滚日志，因此无法进行回滚操作。会保留表的结构定义。使用 TRUNCATE 命令时，必须对表具有 TRUNCATE 权限。

语法如下：

TRUNCATE TABLE table\_name;

（2）DELETE：该命令也可以用于删除表中的所有数据，并且可以产生回滚日志，因此可以进行回滚操作。使用 DELETE 命令时，必须对表具有 DELETE 权限。

语法如下：

DELETE FROM table\_name;

TRUNCATE 和 DELETE 命令在处理数据库中的数据时有着不同的底层实现原理。TRUNCATE 命令是通过直接删除表中的数据来完成操作的，这个过程会释放表所占用的空间，并将表重置为其最初的状态，因此 TRUNCATE 命令通常比 DELETE 命令更快速。

但是，TRUNCATE 命令不仅无法撤销，而且也不触发表上的 DELETE 触发器。与之相反，DELETE 命令是通过逐行删除数据来完成的，这意味着 DELETE 命令可以在事务内回滚，并且会触发表上的 DELETE 触发器。但是，DELETE 命令对于大量的数据可能会影响性能。

因此，在选择使用 TRUNCATE 还是 DELETE 命令时，需要根据实际情况综合考虑。如果只需要快速地删除整个表中的数据并且不需要触发任何触发器，则 TRUNCATE 命令可能是更好的选择。但是，如果需要撤销操作或需要触发触发器，则 DELETE 命令可能是更安全和可靠的选择。

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，可以考虑在该字段上创建一个索引。使用索引可以大大加快查询速度，因为它允许数据库引擎快速地找到符合条件的记录。

具体来说，可以通过以下几个步骤来创建索引：

1.使用 CREATE INDEX 命令创建索引，指定要在哪个表的哪个字段上创建索引。

2.当插入、更新或删除数据时，数据库会自动更新索引，以确保其与实际数据保持同步。

3.在查询时，通过 WHERE 子句指定要查询的字段，并使用比较运算符（如 <）来筛选出符合条件的记录。

举个例子，如果有一个包含客户订单信息的表，其中有一个名为 order\_amount 的字段，经常需要查询订单金额小于1000的信息，那么可以在 order\_amount 字段上创建一个索引，如下所示：

CREATE INDEX idx\_order\_amount ON customer\_orders (order\_amount);

然后，在查询时，可以使用以下语句来检索金额小于1000的订单信息：

SELECT \* FROM customer\_orders WHERE order\_amount < 1000;

这样就可以大大提高查询效率，并且避免了全表扫描的情况。当然，需要注意的是，对于小表，创建索引可能并不会带来明显的性能提升，反而会增加数据库的存储和维护成本。因此，在实际应用中，需要根据表的大小、查询频率等因素综合考虑是否需要创建索引。

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：物化视图是一种预先计算并存储计算结果的视图。它可以在查询中直接引用，而不需要每次都重新计算视图的结果。物化视图通常用于需要经常重复运算、耗时较长的查询场景，以提高查询性能和响应速度。

物化视图与普通视图（也称为虚拟视图）的主要区别在于它们的计算方式和存储方式：

1. 计算方式：普通视图是通过对基本表进行查询来计算视图的结果。每次查询视图时，数据库都会重新执行相应的 SELECT 语句以获取最新的结果。而物化视图事先将结果计算出来，并存储在磁盘上。这意味着在查询物化视图时，数据库无需重新计算视图的结果，只需要读取存储在磁盘上的数据即可。
2. 存储方式：普通视图不存储任何数据，只是一个虚拟表结构。而物化视图是实际存储在磁盘上的物理表。因此，物化视图需要占用更多的存储空间，但却可以提供更快的查询效率。

在实际应用中，可以使用物化视图来优化以下场景：

1. 频繁执行的复杂查询：当某个查询需要执行多个 JOIN 操作、聚合操作和排序操作时，可以使用物化视图将计算结果缓存起来。这样，每次查询时只需要读取缓存数据即可。
2. 支持高并发的 OLAP 系统：在 OLAP（联机分析处理）系统中，用户经常需要根据大量的历史数据进行复杂的查询和分析。使用物化视图可以极大地提升系统的响应速度和并发能力。
3. 缓存一些常用的报表数据：当需要生成一些常用的报表或者统计数据时，可以使用物化视图先将其预先计算出来，并存储在磁盘上。这样，每次生成报表时，只需要从物化视图中读取数据即可，不需要重新计算。

需要注意的是，物化视图虽然可以提高查询性能，但也会增加数据库的存储和维护成本。因此，在使用物化视图时需要综合考虑各种因素，并根据实际情况进行选择。

1. **学校模型ER图绘制**

